

# DIPLOMARBEIT

# NEUORGANISATION UND AUTOMATISIERUNG DES WORKFLOWS DER INSTRUMENTELLEN NEUTRONENAKTIVIERUNGSANALYSE ZUR ERHÖHUNG DER REPRODUZIERBARKEIT UND DATENSICHERHEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur (Dipl.-Ing./DI)**

im Rahmen des Studiums

**Technische Physik**

eingereicht von

**BENJAMIN HINDINGER, BSc**

Matrikelnummer 01063530

Studienkennzahl E 066 461

ausgeführt am Atominstitut  
der Fakultät für Physik der Technischen Universität Wien

Betreuerin: Ao.Univ.Prof.Dipl.-Ing.Dr.techn. Christina Strelli

Mitwirkung: Senior Scientist DI.Dr.techn. Johannes Sterba

Wien, 2. Mai 2018

---

Verfasser

---

Betreuerin

## Zusammenfassung

In dieser Arbeit wird der Messaufbau zur instrumentellen Neutronenaktivierungsanalyse (INAA) am Atominstitut (ATI) der Technischen Universität Wien im Rahmen einer Diplomarbeit systematisch neu organisiert und optimiert.

Die weiteren Komponenten der INAA, welche nicht der Messung angehören, sind die Analyse der gemessenen Spektren und die Auswertung der Energielinien bis zu den resultierenden Konzentrationen der Elemente, welche durch die INAA bestimmt werden sollen.

Diese Arbeitsschritte werden im Zuge der Neuorganisation des Messsystems ebenfalls optimiert. Durch die Erstellung eines benutzerfreundlichen Programms in Visual Basic werden die Komponenten in einem Mess-, Analyse- und Auswertungsprogramm vereint und einige nützliche Funktionen zur Überprüfung der Prozesse und Ergebnisse implementiert. Ziele sind unter anderem die Vermeidung systematischer Fehler und Verkleinerung des Fehlers der resultierenden Nuklidkonzentrationen.

Die obersten Direktiven sind die Sicherstellung der Reproduzierbarkeit der resultierenden Daten und Ergebnisse und die Erhöhung der Datensicherheit. Dies soll hauptsächlich durch umfassende automatische Dokumentation und der Minimierung von manuellen Eingriffen in den Prozess der INAA erfolgen.

Für unvermeidbare Manipulationen des Prozesses sollen automatische und manuelle Dokumentationsmöglichkeiten implementiert werden, um die Wissenschaftlichkeit, aus den Ergebnissen hervorgehender Studien, zu gewährleisten.

*Stichwörter:* Instrumentelle Neutronenaktivierungsanalyse, INAA, Reproduzierbare Forschung, Gammaskopie

## Abstract

In this diploma thesis the system for measurement as part of the Instrumental-Neutron-Activation-Analysis (INAA) at the Atominstitut (ATI) at TU Wien should systematically be renewed and optimized.

Components of the INAA, which are not counted among the measurement, are the analysis of measured spectra and evaluation of the containing peaks. This leads to the determination of the concentrations of elements analyzed by use of the INAA.

These steps will be optimized beside the reorganization of the measurement system. The three main components will be united due to the creation of a user-friendly computer program with Visual-Basic. Furthermore several useful routines for supervising the processes will be added. Aims of the reorganization are to avoid systematical errors and to reduce the standard error of the resulting nuclide concentrations.

The main objectives are to ensure the reproducibility of the resulting data and outcomes and enhancement of data integrity.

These aims should be achieved with broad automatic documentation routines and by minimizing manipulations of the INAA process by users.

Some manipulations are unavoidable, therefore more automated and manual documentation possibilities have to be implemented to increase and guarantee the rigor of science with regard to emerging papers.

*Keywords:* Instrumental neutron activation analysis, INAA, Reproducible Research, Gamma spectroscopy

## Inhaltsverzeichnis

1. Einleitung und Motivation.....	1
2. Theoretischer Überblick.....	3
2.1. Instrumentelle Neutronenaktivierungsanalyse .....	3
2.1.1. Allgemeines .....	3
2.1.1.1. Anwendung und Vorteile .....	3
2.1.2. Ablauf .....	4
2.1.2.1. Probenvorbereitung.....	4
2.1.2.2. Aktivierung .....	4
2.1.2.3. Messung.....	6
2.1.2.4. Analyse .....	8
2.1.2.5. Auswertung .....	9
2.2. Reproduzierbare Forschung .....	12
2.2.1. Allgemeines .....	12
2.2.2. Problematik.....	12
2.2.3. Methoden und Aufgaben.....	14
2.2.4. Ableitungen.....	14
3. Neuorganisation des Messaufbaus.....	17
3.1. Vorbereitungen .....	17
3.1.1. IST-Zustand .....	17
3.1.2. SOLL-Zustand .....	17
3.1.3. Problematik.....	17
3.2. Durchführung.....	18
3.2.1. Updates .....	18
3.2.2. Probenwechsler .....	18
3.2.3. Neuaufbau Messsystem-Russe .....	20
4. Erstellung des Programms - DUAL_LFC.EXE .....	23
4.1. Vorbereitungen .....	24
4.1.1. Genie 2000 .....	24
4.1.2. Visual Basic(VB) - .NET Framework .....	25
4.1.3. Reproduzierbarkeit.....	26
4.2. Klasse - DLFC_index.vb .....	26
4.2.1. Problematik der Reproduzierbarkeit - Stammklasse.....	26
4.2.2. Implementierung der Stammklasse .....	28
4.2.2.1. Öffentliche Funktionen.....	28
4.2.2.2. Allgemeine Funktionen .....	29

4.2.2.3.	Fehlerbehandlung in der Stammklasse .....	30
4.2.3.	Tochterklassen von DLFC_index.vb .....	31
4.2.3.1.	Klasse - auto_auswahl.vb.....	31
4.2.3.2.	Klasse - control_box.vb .....	31
4.3.	Klasse - Messung .....	32
4.3.1.	Problematik der Reproduzierbarkeit - Messklasse .....	32
4.3.2.	Implementierung der Messklasse.....	33
4.3.2.1.	Fehlerbehandlung in der Messklasse .....	35
4.4.	Klasse - Analyse.vb .....	36
4.4.1.	Problematik der Reproduzierbarkeit - Analyseklasse .....	36
4.4.2.	Implementierung der Analyseklasse.....	37
4.4.2.1.	Kalibrierung .....	37
4.4.2.2.	Standardanalyse .....	38
4.4.2.3.	Spezialanalyse .....	39
4.4.2.4.	Interactive-Peak-Fit (IPF) .....	43
4.4.2.5.	Allgemeine Funktionen .....	44
4.4.2.6.	Fehlerbehandlung in der Analyseklasse .....	45
4.4.3.	Tochterklasse von Analyse.vb.....	46
4.5.	Klasse - Auswertung.vb .....	46
4.5.1.	Problematik der Reproduzierbarkeit - Auswertungsklasse .....	46
4.5.2.	Implementierung der Auswertungsklasse.....	47
4.5.2.1.	Einstiegsfunktion - auswerten() .....	49
4.5.2.2.	Erste Berechnung.....	51
4.5.2.3.	Standards.....	52
4.5.2.4.	Ergebnis.....	53
4.5.2.5.	Einlesefunktionen .....	53
4.5.2.6.	Ausgabefunktionen.....	56
4.5.2.7.	Fehlerbehandlung in der Auswertungsklasse .....	57
4.5.3.	Tochterklasse von Auswertung.vb .....	57
5.	Überprüfung der Ergebnisse.....	60
5.1.	Überprüfung der Messung .....	60
5.2.	Überprüfung der Analyse.....	60
5.3.	Überprüfung der Auswertung.....	64
6.	Bedienungsanleitung .....	71
6.1.	Anwendung .....	71
6.1.1.	Ein- und Ausgabedateien.....	71

6.1.1.1.	CAM-Dateien (.CNF) .....	72
6.1.1.2.	Kalibrierdateien (.CAL) .....	72
6.1.1.3.	Analysesequenzen (.ASF) .....	72
6.1.1.4.	Textdatei (.TXT) .....	72
6.1.1.5.	Reportdateien (.RPT) .....	73
6.1.1.6.	Vorlagedateien (.TPL) .....	73
6.1.1.7.	Textdateien (.CSV) .....	73
6.1.2.	Messung .....	74
6.1.2.1.	Kalibrierung .....	74
6.1.2.2.	Benutzereingaben - Messung .....	75
6.1.2.3.	Starten der Messung .....	76
6.1.2.4.	Fehlerquellen und -meldungen der Messung .....	76
6.1.3.	Kalibrierung .....	77
6.1.3.1.	Benutzereingaben - Kalibrierung .....	77
6.1.3.2.	Fehlermeldungen .....	77
6.1.4.	Standardanalyse .....	78
6.1.4.1.	Benutzereingaben - Standardanalyse .....	78
6.1.4.2.	Individuelle Analysesequenz .....	78
6.1.4.3.	Durchführung der Standardanalyse .....	79
6.1.4.4.	Fehlerquellen und -meldungen der Standardanalyse .....	79
6.1.5.	Spezialanalyse .....	80
6.1.5.1.	Benutzereingaben - Spezialanalyse .....	80
6.1.5.2.	Anpassung einer Analyseparameter-Datei .....	80
6.1.5.3.	Durchführung der Spezialanalyse .....	81
6.1.5.4.	Fehlerquellen und -meldungen der Spezialanalyse .....	81
6.1.6.	Interactive-Peak-Fit .....	82
6.1.6.1.	Durchführung des IPF .....	82
6.1.6.2.	Fehlerquellen und -meldung des IPF .....	83
6.1.7.	Auswertung .....	83
6.1.7.1.	Benutzereingaben - Auswertung .....	84
6.1.7.2.	Erstellung der Probendatei .....	84
6.1.7.3.	Anpassung der Nukliddateien .....	84
6.1.7.4.	Durchführung der Auswertung .....	85
6.1.7.5.	Fehlerquellen und -meldungen der Auswertung .....	87
6.1.8.	Automatische Mess- /Analyse- und Auswertungssequenz .....	88
6.1.8.1.	Benutzereingaben - AUTO-Sequenz .....	88

6.1.8.2.	Durchführung der AUTO-Sequenz.....	89
6.1.8.3.	Fehlerquellen und -meldungen der AUTO-Sequenz.....	89
6.1.9.	Referenzen anzeigen.....	90
6.2.	Implementierung.....	90
6.2.1.	DUAL_LFC - Setup.....	90
6.2.2.	Modifizierung von Programmteilen.....	91
6.2.2.1.	Allgemeine Veränderungen.....	91
6.2.2.2.	COM-DLL's.....	92
6.2.2.3.	BATCH-Befehle.....	92
6.3.	Kurzanleitung.....	94
6.3.1.	Messung.....	94
6.3.2.	Kalibrierung.....	94
6.3.3.	Standardanalyse.....	94
6.3.4.	Spezialanalyse.....	94
6.3.5.	IPF.....	94
6.3.6.	Auswertung.....	94
7.	Conclusio.....	95
8.	Literatur.....	98
9.	Abbildungs- und Tabellenverzeichnis.....	100
ANHANG A - Canberra COM-DLL Syntax.....		101
ANHANG B - Quellcode DLFC_ADV (Probenwechsler).....		101
ANHANG C - Quellcode DUAL_LFC.....		102
Klasse - DLFC_index.vb.....		102
Tochterklasse - auto_auswahl.vb.....		109
Tochterklasse - control_box.vb.....		110
Klasse - messung.vb.....		110
Klasse - analyse.vb.....		113
Tochterklasse - input_box.vb.....		120
Klasse - auswertung.vb.....		120
Tochterklasse - standards.vb.....		137

# 1. EINLEITUNG UND MOTIVATION

Ziel dieser Diplomarbeit ist es, den aktuellen Messaufbau zur instrumentellen Neutronenaktivierungsanalyse (INAA) zu modernisieren und den darauf aufbauenden Messvorgang sowie die Analyse- und Auswertung der Messergebnisse zu optimieren und wie bereits von Westphal 2007 [13] vorgeschlagen, weitestgehend zu automatisieren.

Die Neuorganisation des Messaufbaus inkludiert den Austausch der veralteten Mess-PC's durch hochwertige Industrierechner von IBM sowie die Sicherstellung der Funktionstüchtigkeit der Schnittstellen zu Messdetektor und Probenwechsler. Durch die Modernisierung der Messrechner ist eine Neuprogrammierung der Messroutine notwendig, da das Vorgängerprogramm nicht für die neuen Rechner geeignet ist.

Als Ersatz zum Vorgängermodell muss eine neue Routine für eine gewöhnliche Analyse der Messergebnisse erstellt werden. In Zuge dessen soll eine größtmöglich automatisierte Analyse- und Auswertungsmethode entwickelt und implementiert werden, welche der meistgenutzten Anwendung angepasst wird.

Für die Auswertung der analysierten Daten wurde bisher ein umfangreiches Excel Dokument verwendet. Die Auswertung der analysierten Daten soll ebenfalls möglichst automatisiert, mittels einer neuen Routine, realisiert werden.

Daraus ergeben sich folgende Teilbereiche in denen diese Diplomarbeit gegliedert wird:

- Aufbau und Modernisierung des Messaufbaus
- Programmierung der Messroutine
- Programmierung der Analysemethoden
- Programmierung der automatisierten Auswertung

Oberste Intention ist es, durch die Automatisierung des Mess- /Analyse- und Auswertungsprozesses die Reproduzierbarkeit dieser Abläufe sicherzustellen. Dazu soll manuelle Manipulation von Daten während des Mess- /Analyse- oder Auswertungsprozesses unterbunden werden oder, wenn unbedingt notwendig, automatisiert dokumentiert werden.

Die Motivation dieser Diplomarbeit ergibt sich einerseits durch die Notwendigkeit der Modernisierung des Messaufbaus zur Minimierung der aktuellen Fehleranfälligkeit sowie der Erhöhung der Geschwindigkeit des Messsystems und Erhöhung der Effizienz durch Teilautomatisierung des Workflows. Andererseits ist die Arbeit motiviert durch die Verbesserung der Reproduzierbarkeit der Neutronenaktivierungsanalyse (NAA) um wissenschaftlich fundierte Daten zu erhalten.

Aktuell wird für die Messung, Analyse und Auswertung jeweils ein eigenes Programm mit unterschiedlicher Struktur verwendet. Um die Reproduzierbarkeit zu festigen soll ein einheitliches Programm erstellt werden, in welchem diese Teilbereiche zusammengeführt werden. Zusätzlich soll in diesem Programm die Fehleranfälligkeit durch außergewöhnliche Proben, sowie falsche und ungünstige Benutzereingaben minimiert werden.

Die Zusammenführung der Teilbereiche eröffnet auch die Möglichkeit die einzelnen Schritte der Messung, Analyse und Auswertung automatisiert nacheinander auszuführen und eine weitere Fehlerquelle, die manuelle Durchführung der Teilbereiche der INAA, zu eliminieren.

Aus diesen Vorgaben leiten sich weitere notwendige Teile dieser Arbeit ab:

- Überprüfung des erstellten Programms
- Bedienungsanleitung

Diese Arbeit dient zur Beschreibung der Erstellung und der Anwendungsmöglichkeiten des Programms. Hauptpriorität ist es, den Anwendern am Atominstitut den Aufbau und die Verwendung des Programms, vollständig und verständlich zu vermitteln.

## 2. THEORETISCHER ÜBERBLICK

### 2.1. INSTRUMENTELLE NEUTRONENAKTIVIERUNGSANALYSE

#### 2.1.1. Allgemeines

Die Arbeit von György Hevesy und Hilde Levi von 1936 [2] gilt als Grundstein für die Neutronenaktivierungsanalyse. Darin aktivierten sie Proben seltener Erden mit einer Ra-Be-Neutronenquelle und bestimmten deren Aktivität mit einem Geiger-Müller-Zählrohr. Häufige Anwendungen und Weiterentwicklungen zeigten sich aber erst mit der Verbreitung von Kern- bzw. Forschungsreaktoren, wodurch passende Quellen thermischer Neutronen hoher Flussdichte zur Verfügung standen [3].

Die instrumentelle Neutronenaktivierungsanalyse (INAA) ist der klassische Teilbereich der Neutronenaktivierungsanalyse, welche es erlaubt, Proben einen chemischen Fingerabdruck zuzuweisen [3]. Die Neutronenaktivierungsanalyse umfasst außerdem die Prompte-Gamma-Aktivierungsanalyse (PGAA) und die Radiochemische Neutronenaktivierungsanalyse (RNAA). Die PGAA unterscheidet sich stark im Messaufbau von der INAA, aufgrund der Messung der sofort ausgesendeten Gammaquanten während der Aktivierung [2]. Die RNAA weist eine aufwendigere Probenaufbereitung wegen der Verwendung von radiochemische Trennmethoden auf [4].

Bei der INAA werden Proben, durch Beschuss mit thermischen Neutronen, aktiviert und die anschließenden erfolgenden radioaktiven Zerfälle analysiert bzw. deren emittierte Gammastrahlung mit einem Gammastrahlendetektors gemessen. [1]

#### 2.1.1.1. Anwendung und Vorteile

Die Anwendungen der INAA sind vielschichtig. Einige davon sind die Qualitätssicherung von Materialien in Industrie und Medizin durch Sicherstellung der Reinheit der Materialien wie z.B. Verunreinigungen in Silizium von Mikroelektronik, geowissenschaftliche Untersuchungen von Meteoriten, Messungen von Umweltproben auf Schwermetallen, sowie kunsthistorische Fragestellungen, welche Spurenelemente in Farbpigmenten vorhanden sind und vielseitige Anwendungen in der Archäometrie, zur Bestimmung von Zusammensetzung und Herkunft von archäologischen Ausgrabungen [5].

Grundsätzlich hängt die Art der Analysemethode natürlich von der Beschaffenheit der Proben, Probenmatrix, nachzuweisende Elemente und Konzentrationen sowie von der allgemeinen Fragestellung ab.

Vorteile der INAA sind im Speziellen,

- dass die chemische Form sowie der physikalische Zustand, in welcher die Proben vorliegen, grundsätzlich keine Rolle spielen[2].

- dass aufwendige Probenvorbereitungen im Gegensatz zur RNAA nicht notwendig sind, wodurch das Risiko von Verunreinigungen wegfällt [2].
- dass häufige Hauptbestandteile von Proben, oben angeführter Anwendungen, schwer zu aktivierende bzw. wegen kurzer Halbwertszeiten zu messende Elemente wie H, C, N, O, Si sind und welche im Gammaskpektrum keinen störenden Hintergrund zur Folge haben [2].
- dass die Messung fast aller Elemente mit Ordnungszahl größer acht, mit Ausnahme von Schwefel und Phosphor, grundsätzlich möglich ist [3].
- dass Spurenelemente mit Genauigkeiten von bis zu  $10^{-9}$  g/kg nachgewiesen werden können [3] [5].
- dass der Nachweis einer größeren Anzahl an Elementen durch eine INAA gleichzeitig möglich ist - Multielementanalyse [6].

## 2.1.2. Ablauf

### 2.1.2.1. Probenvorbereitung

Kleine Mengen der Proben werden zu einem Pulver zermahlen und in Phiolen aus Suprasil™ - Quarzglas eingewogen. Üblich sind Einwaagen von 100 - 200 mg. Die Glasphiolen werden verschweißt und anschließend in zylinderförmigen Kunststoff-Probenbehältern, welche für die Verwendung im Probenwechsler geeignet sind, fixiert. Die Probenbehälter werden nummeriert und eine komplette Liste der Probenreihe inklusive Bezeichnungen und Einwaagen angefertigt [1] [3]. In Abbildung 1 ist eine Probe eingewogen und eingeschweißt in einer Suprasil-Glasphiole zu sehen.



**Abb.1:** In Suprasil-Glasphiole eingeschweißte Probe (Quelle: HMI Berlin, <http://portal.uni-freiburg.de/janiak/Lehre/ac-pdf/vl22>)

### 2.1.2.2. Aktivierung

Aktiviert werden in der Regel Proben stabiler Atomkerne durch Beschuss von thermischen Neutronen. Kernreaktionen wie Neutroneneinfang ( $n, \gamma$ ) wandeln stabile Nuklide in instabile Radionuklide um. In Abbildung 2 ist der Prozess der Aktivierung eines Atomkerns ersichtlich. Darin wird durch Neutronenbeschuss ein stabiler Atomkern  ${}^A_ZX$  in ein Isotop des gleichen Elements umgewandelt, welches sich in einem angeregten Zustand befindet und durch

Emission eines bei der PGAA gemessenen Gammaquants, in einen niederenergetischeren Zustand übergeht. Daraus wird ersichtlich, warum die Detektion des prompt emittiertem Gammaquants im Reaktor während der Aktivierung erfolgen muss.

Durch anschließenden Betazerfall und Emission eines weiteren, bei der INAA gemessenen Gammaquants geht der Atomkern in ein anderes stabiles Element über [2]. Das instabile Nuklid, welches das verzögerte Gammaquant emittiert, kann Halbwertszeiten von Sekunden bis viele Jahre aufweisen [3].

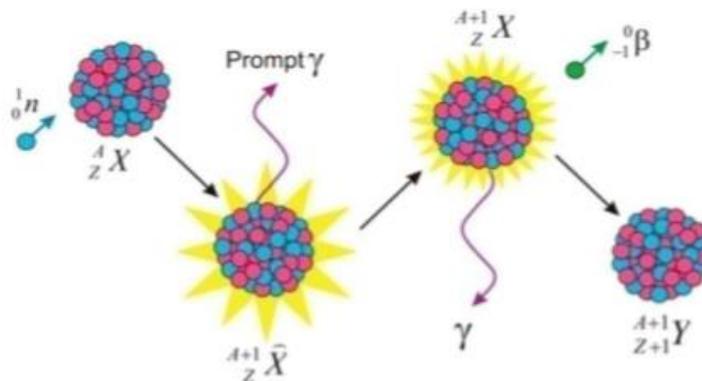


Abb.2: Aktivierung stabiler Atomkerne [3]

Gammastrahlung tritt oft in Verbindung mit Alpha- oder Betastrahlung auf, da sich Atomkerne nach derartigen Zerfällen oft in angeregte Zustände befinden [7]. Ein angeregter Zustand, welcher die Emission eines Gammaquants zur Folge hat, kann aber ebenso durch andere Kernreaktionen, wie z.B. Neutroneneinfang oder Absorption eines Gammaquants erzeugt werden [8].

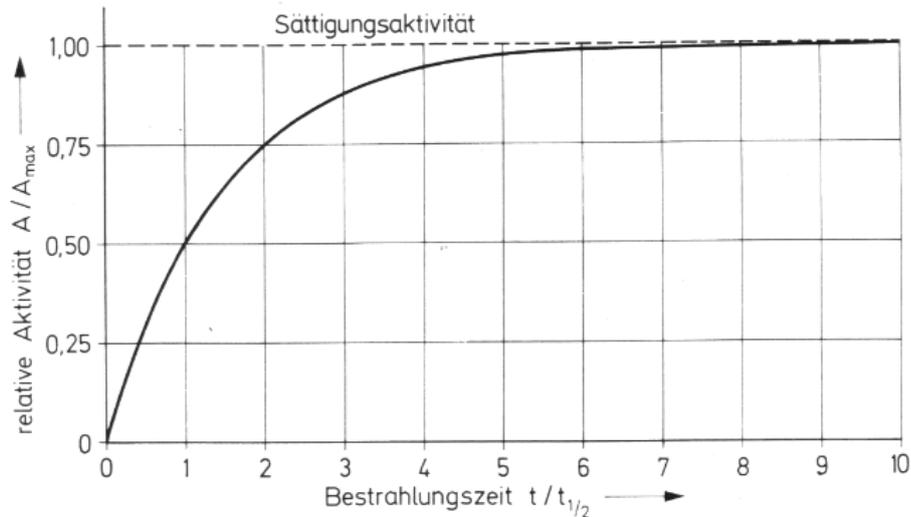
Dabei haben Atomkerne in angeregten Zuständen höhere Rotations- und Schwingungsfreiheitsgrade als Kerne im Grundzustand.

Die Energie des emittierten Gammaquants ergibt sich aus der Energiedifferenz zwischen angeregtem Zustand  $E_k$  und niederenergetischerem Zustand  $E_i$  zu  $h \cdot \nu = E_k - E_i$  und liegt im Bereich von  $10^4$ - $10^7$  eV [7].

Das Energiespektrum emittierter Gammastrahlen setzt sich demnach aus diskreten Energielinien zusammen. Da diese Energien für Radionuklide charakteristisch sind und meist mehrere Energielinien zu gewissen Wahrscheinlichkeiten auftreten, können die ursächlichen Radionuklide durch die nachfolgend beschriebene Messung identifiziert werden [12].

In dieser Arbeit wurde die Aktivierung mit thermischen Neutronen der Flussdichte  $10^{13} \frac{1}{\text{cm}^2 \cdot \text{s}}$ , am Forschungsreaktor des Atominstututs der TU Wien - vom Typ TRIGA Mark-II - durchgeführt [1] [3].

Die richtige Bestrahlungsdauer und Zeitpunkte der Messungen sind wesentliche Parameter bei der INAA. Die Aktivität einer Probe in Abhängigkeit von der Bestrahlungsdauer ist in Abbildung 3 dargestellt.

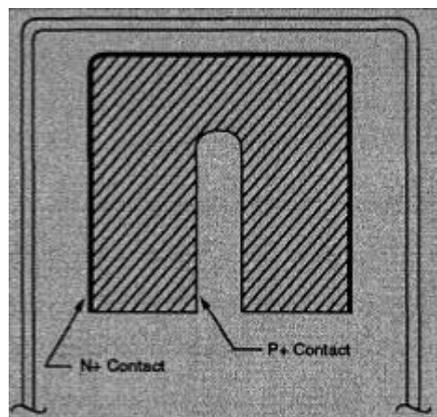


**Abb.3:** Anstieg der relativen Aktivität von Proben während der Bestrahlung [6]

Man erkennt, dass die Aktivität einer Sättigungsaktivität entgegenstrebt. Dieser Verlauf ist für alle Nuklide gleich, wobei bei einer Bestrahlungszeit von doppelter Halbwertszeit bereits 75 % der Sättigungsaktivität erreicht ist [6]. Für die übliche Anwendung am ATI wird eine Bestrahlungsdauer von 30-40 Stunden definiert.

### 2.1.2.3. Messung

Das Gammaskpektrum wird meist mittels Halbleiterdetektoren wie zum Beispiel eines „coaxial-high-purity -Ge-Detektor“ (HPGe) aufgenommen. HPGe enthalten als Detektorkristall hochreines Germanium. Dieser fungiert als Diode, dotiert mit Lithium(Donor) und Bor(Akzeptor), welche an eine Hochspannung in Sperrichtung angelegt ist und zur Detektion von Gammaquanten im Bereich von ca. 50 keV bis 3 MeV mit guter Energieauflösung und hohen Zählraten dient. Zur Unterdrückung von Hintergrundrauschen wird eine dicke Abschirmung aus altem Blei und eine Kühlung mit flüssigem Stickstoff benötigt. In Abbildung 4 ist ein generelles Schnittbild des Messkristalls eines HPGe dargestellt [9] [10].



**Abb.4:** Schnittbild - coaxial high-purity -Ge-Detektor [9]

Gammastrahlung erzeugt im Detektorkristall Elektron-Loch-Paare, welche aufgrund des elektrischen Feldes zu Elektroden wandern und einen Stromimpuls erzeugen, der in anschließender Messelektronik registriert wird.

Die Messelektronik dient zur Aufbereitung des Signals und korrekter Registrierung des Impulses. Während ein Impuls registriert wird, kann kein weiteres Signal von der Messelektronik angenommen werden. Diese Zeit wird als Totzeit bezeichnet [9] [11].

Das Detektorsystem, welches in dieser Arbeit beschrieben wird, beinhaltet aus diesem Grund ein System zur Zählratenkorrektur - Loss-Free-Counting (LFC), welche mittels Korrekturwerte die Totzeit berücksichtigt und die verminderte Zählrate an die wahre Zählrate annähert. Im Wesentlichen ist die LFC-Korrektur vor allem bei großen und variablen Zählraten aufgrund langer Totzeiten notwendig und sinnvoll [12]. Die Korrekturmethode LFC und deren Funktionsweise sind in den Arbeiten [9-12] näher ausgeführt.

Weiters ist der für die Messung von Probenreihen verwendete Detektor mit einem Probenwechslersystem gekoppelt. Dies ist ein pneumatisch gesteuertes, einfaches aber effektives System zur automatisierten Messung von Probenreihen, praktisch unbeschränkter Probenanzahl. Dazu werden Proben durch Druckluft, welche von einem Industriekompressor erzeugt wird, über ein System von Kunststoffschläuchen und Rohren in die Messkammer des Detektors injiziert beziehungsweise nach erfolgter Messung aus dem Detektor extrahiert, bevor die nächste Probe eingeschossen wird. Der Druck beträgt ca. 5 bar und die Transportzeit in oder aus der Messkammer des Detektors einige Sekunden. Der Probenwechsler kann manuell oder mit entsprechender Routine auch durch den Mess-PC gesteuert werden. Durch den Probenwechsler ergibt sich eine fixe Probengeometrie für den Detektor, ersichtlich in Abbildung 5. Die Probe ist im Hintergrund im Transportrohr des Probenwechslers ersichtlich.



**Abb.5:** Probengeometrie des verwendeten Messdetektors [Bild des Verfassers]

Das verwendete duale Messsystem speichert gleichzeitig 8K Kanäle des korrigierten, sowie 8K Kanäle des unkorrigierten Spektrums in einem einzigen 16K Spektrum ab. Für diese Anwendung finden zwei Messungen statt. Die erste - „mittel“ - mit einer Abklingzeit ab der Aktivierung von 5 Tagen und die zweite - „lang“ - mit einer Abklingzeit ab der Aktivierung von 28

Tagen. Die Messzeitpunkte und Halbwertszeiten der Radionuklide, welche analysiert werden sollen, müssen dementsprechend aufeinander angepasst werden.

#### 2.1.2.4. Analyse

Es müssen eine Reihe von Nukliden definiert werden, deren Konzentrationen während der Auswertung berechnet werden sollen. Wie bereits erwähnt bieten sich für die INAA die Analyse von Spurenelementen in ansonsten reinen Materialien an.

Dementsprechend werden Nuklide gewählt, welche durch den Aktivierungsprozess Radionuklide erzeugen, die den folgenden Kriterien entsprechen:

- Die Halbwertszeiten der Radionuklide müssen mittel bis langfristig sein, um mit zwei generalisierte Messungen (nach 5 und 28 Tagen) entsprechende Zerfälle messen zu können[1].
- Radionuklide zerfallen mit unterschiedlich wahrscheinlichen Zerfallskanälen. Mindestens einer der resultierenden Linien sollte frei von Interferenz anderer Energielinien sein[15].
- Bieten sich für ein Radionuklid mehrere Energielinien an, sollte die intensivste zur Analyse herangezogen werden, da das Signal-zu-Hintergrund-Verhältnis (signal-to-noise-ratio - SNR) günstiger ist[15].
- Es sollten viele Linien verteilt über das gesamte Spektrum analysiert werden, um die Güte der Analyse zu erhöhen. Zu Kontrollzwecken können auch mehrere Linien desselben Radionuklids analysiert werden. [15].

Die Elemente deren Konzentrationen bestimmt werden sollen, die dafür eingesetzten Radionuklide, deren Ordnungszahlen (Z) und Halbwertszeiten (HWZ) sowie die neun gewählten Energielinien, sind für die mittlere Messung nach Energien sortiert dargestellt und in anschließender Tabelle ersichtlich. Für Natrium werden zwei Energielinien analysiert.

Elemente	Radionuklid	Z	HWZ [s]	Energie [keV]
Samarium	<sup>153</sup> Sm	62	166572	103,2
Lutetium	<sup>177</sup> Lu	71	581818	208,4
Uran	<sup>239</sup> Np (Neptunium)	93	203602	277,4
Arsen	<sup>76</sup> As	33	93121,2	559,1
Wolfram	<sup>187</sup> W	74	86040	685,7
Natrium*	<sup>24</sup> Na	11	53852	1369
Kalium	<sup>42</sup> K	19	44496	1525
Lanthan	<sup>140</sup> La	57	144986	1596
Natrium	<sup>24</sup> Na	11	53852	2754

**Tab.1:** Auswahl der Nuklide und Energien der mittleren Messung; \* zu Kontrollzwecken

Die Radionuklide, deren Halbwertszeiten und deren 22 gewählte Energielinien sind für die lange Messung nach Energien sortiert und in folgender Tabelle ersichtlich. Für Tantal werden zwei Energielinien analysiert.

Element	Nuklid	Z	HWZ [s]	Energie [keV]
Cer	<sup>141</sup> Ce	58	2808090	145,4
Ytterbium	<sup>169</sup> Yb	70	2767046	177,2
Lutetium*	<sup>177</sup> Lu	71	581818	208,4
Thorium	<sup>233</sup> Pa (Protactinium)	91	2329950	312,2
Chrom	<sup>51</sup> Cr	24	2393450	320,1
Hafnium	<sup>181</sup> Hf	72	3662500	482,2
Barium	<sup>131</sup> Ba	56	993600	496,3
Strontium	<sup>85</sup> Sr	38	5593017	513,99
Neodym	<sup>147</sup> Nd	60	948672	531
Zirkonium	<sup>95</sup> Zr	40	5531330	756,7
Cäsium	<sup>134</sup> Cs	55	65164700	795,9
Nickel	<sup>58</sup> Co (Kobalt)	28	6114960	810
Terbium	<sup>160</sup> Tb	65	6246720	879,4
Rubidium	<sup>86</sup> Rb	37	1609720	1077
Eisen	<sup>59</sup> Fe	26	3845060	1099,2
Zink	<sup>65</sup> Zn	30	21104100	1115,5
Scandium	<sup>46</sup> Sc	21	7239460	1120,5
Kobalt	<sup>60</sup> Co	27	1,66 * 10 <sup>8</sup>	1173,2
Tantal*	<sup>182</sup> Ta	73	9886750	1189
Tantal	<sup>182</sup> Ta	73	9886750	1221,3
Europium	<sup>152</sup> Eu	63	4,27 * 10 <sup>8</sup>	1408
Antimon	<sup>124</sup> Sb	51	5201280	1691

**Tab.2:** Auswahl der Nuklide und Energien der langen Messung; \* zu Kontrollzwecken

### 2.1.2.5. Auswertung

Werden gleichzeitig mit den zu analysierenden Proben Referenzstandards mit bekannten Konzentrationen der zu bestimmenden Elemente im Reaktor aktiviert, kann die Konzentration einer Probe  $K_{\text{Probe}}$  im Prinzip aus den Zählraten der Messungen und der bekannten Konzentration des Standards  $K_{\text{Standard}}$  mit (1) bestimmt werden [15].

$$\frac{\text{Zählrate}_{\text{Probe}}}{K_{\text{Probe}}} = \frac{\text{Zählrate}_{\text{Standard}}}{K_{\text{Standard}}} \rightarrow K_{\text{Probe}} = \frac{\text{Zählrate}_{\text{Probe}}}{\text{Zählrate}_{\text{Standard}}} * K_{\text{Standard}} \quad (1)$$

Die Herleitung der Relation (1) aus der Aktivierungsgleichung (2) ist in [3], [6] und [14] durchgeführt.

$$A = N_0 \Phi \sigma (1 - e^{-\lambda t_{ir}}) \quad \text{mit } \lambda = \frac{\ln 2}{t_{1/2}} \quad (2)$$

Wobei

$N_0$	Menge des stabilen Probenelements
$\Phi$	Neutronenfluss im Reaktor
$\sigma$	Wirkungsquerschnitt der Neutronen
$t_{ir}$	Bestrahlungszeit der Probe
$\lambda$	Zerfallskonstante
$t_{1/2}$	Halbwertszeit

Die Berechnungen der Auswertung werden in wenigen Schritten wie folgt eingeteilt:

Zuerst wird mittels (3) die Zählrate pro Gramm Probeneinwaage, bezeichnet als „Erste Berechnung“ für jeden Peak berechnet, worin die gemessenen Zählraten mit Termen für die Abklingzeit und die Messdauer der Proben korrigiert werden.

$$\text{Zählrate} \left[ \frac{1}{s \cdot g} \right] = \frac{\frac{Cts \cdot \lambda}{e^{-\lambda t_{ab}} \cdot (1 - e^{-\lambda t_{mess}})}}{m} \quad (3)$$

Wobei:

$Cts$	Peakfläche in counts per second
$t_{ab}$	Abklingzeit in Sekunden
$t_{mess}$	Messdauer in Sekunden
$\lambda$	Zerfallskonstante
$m$	Masse der Probeneinwaage

Die Peakfläche wird von der LFC - Reportdatei extrahiert, da bei dieser Peakfläche die Totzeit des Detektorsystems berücksichtigt ist.

Da mit einem Standard nicht alle Elemente mit gleich hoher Qualität bestimmt werden können, beziehungsweise Peaks mancher Nuklide schwerer registriert und gemessen werden können, werden mehrere Standards verwendet. Im zweiten Schritt der Auswertung werden für alle Standards der Probenreihe die Zählraten pro Milligramm Probeneinwaage pro Mikrogramm Element in Gramm Referenzstandard, kurz als R-Wert bezeichnet, mit (4) berechnet.

$$R - \text{Wert} = \frac{\text{Zählrate}_{\text{Standard}}}{K_{\text{Standard}}} \left[ \frac{1}{s \cdot \mu g} \right] \quad (4)$$

Diese R-Werte müssen manuell vom Benutzer (Kapitel 6.1.7.4) oder automatisch überprüft und bearbeitet werden, damit die Standards für die entsprechenden Elemente verwendet werden, deren R-Werte gut übereinstimmen. Meist werden Standards entfernt, welche vom entsprechenden Element relativ wenig enthalten und aufgrund niedrig ausgeprägter Peaks vor dem Messhintergrund die Messung erschwert wird. Aus den erhaltenen R-Werten werden Mittelwerte und Standardabweichungen berechnet. Ist die Standardabweichung der R-Werte höher einem vorher bestimmten Wert, wird sukzessive der am weitesten vom Mittelwert abweichende R-Wert entfernt, bis die Standardabweichung der gewünschten Genauigkeit entspricht [15]. In dieser Arbeit wird als Kriterium eine Standardabweichung kleiner 5 %

bestimmt, wobei bei einer automatischen Bearbeitung das Kriterium weiter gestaffelt wird (Seite 51).

Der Mittelwert, wird wie üblich mit

$$\text{Mittelwert } \bar{x} = \frac{\sum x_i}{n} \quad (5)$$

berechnet, wobei  $x_i$  die R-Werte und  $n$  die Anzahl der R-Werte sind.

Die Standardabweichung  $s$  hingegen, wird aus der Standardabweichung der Stichprobe geschätzt.

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \quad (6)$$

Um den Fehler der Konzentrationen der Nuklide in den Standards zu berücksichtigen, wird der Standardfehler des Mittelwertes ( $SE_{\bar{x}}$  - standard error) mit (7) berechnet.

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}} \quad (7)$$

Für die weitere Auswertung wird der Standardfehler des Mittelwerts durch Division durch den Mittelwert in Prozent umgerechnet und quadriert. Dadurch erhält man ein prozentuelles Fehlerquadrat. Dies ermöglicht die Addition des Fehlers des Mittelwerts zum Fehler der Peakflächenintegration.

Abschließend wird in (3)  $\frac{\text{Zählrate}_{\text{Standard}}}{K_{\text{Standard}}}$  durch den Mittelwert der R-Werte ersetzt und  $K_{\text{Probe}}$  mit (1) berechnet.

Für die Berechnung der Abweichung der Konzentration wird der Fehler aus dem unkorrigierten UNC-Spektrum entnommen, da dieser Fehler aufgrund der kleinen unabhängigen Wahrscheinlichkeiten jedenfalls aus einem Poisson-Prozess hervorgeht. Im Gegensatz dazu wird der Poisson-Prozess durch die LFC-Korrektur verfälscht. Der Fehler des LFC-Spektrums ist daher nicht aussagekräftig.[13]

Der Fehler des unkorrigierten Spektrums muss aufgrund der Korrektur des LFC-Systems im Nachhinein korrigiert werden. Dazu wird mit (9) der unkorrigierte Fehler minimal angeglichen, wobei  $A_{\text{LFC}}$  und  $A_{\text{UNC}}$  die Peakflächen der korrigierten und unkorrigierten Spektren sind und somit  $\frac{A_{\text{LFC}}}{A_{\text{UNC}}}$  als mittlerer Korrekturfaktor dient. [13] Aus

$$\sigma_{\text{korr}}^2 = \sigma_{\text{UNC}}^2 + \frac{1}{A_{\text{UNC}}} * \sigma_W^2 \quad \text{wobei } \sigma_W = \sqrt{\frac{A_{\text{LFC}}}{A_{\text{UNC}}}} \quad (8)$$

folgt

$$\sigma_{\text{korr}} = \sqrt{\sigma_{\text{UNC}}^2 + \frac{A_{\text{LFC}}}{A_{\text{UNC}}^2}} \quad (9)$$

Mit (10) wird der korrigierte Fehler aus (9) quadriert um zu dem prozentuellen Fehlerquadrat des Standardfehlers des Mittelwertes addiert werden zu können.

$$\sigma = \sqrt{\left(\frac{SE_{\bar{x}}}{\bar{x}}\right)^2 + \left(\frac{\sigma_{korrr}}{100}\right)^2} \quad (10)$$

Daraus ergibt sich ein adäquater relativer Fehler  $\sigma$ . Dieser wird für weitere Auswertungen als absoluter Fehler in der Ergebnisdatei *PRN\_auswertung.csv* zur Verfügung gestellt.

## 2.2. REPRODUZIERBARE FORSCHUNG

Zuerst wird das Thema reproduzierbare Forschung unter den Gesichtspunkten der Problematik, Methoden zur Steigerung und Vorgehensweise ganz allgemein behandelt. Anschließend werden aus der Abhandlung individuelle Schlussfolgerungen gezogen, wie Reproduzierbarkeit für diese Arbeit erreicht werden kann.

### 2.2.1. Allgemeines

Das Thema Reproduzierbarkeit von Experimenten ist vermutlich so alt wie wissenschaftliches Arbeiten selbst, da Reproduzierbarkeit eng verknüpft ist mit Definitionen nützlicher Wissenschaft.

Sir Francis Bacon (1620) und Sir Isaac Newton (1726) beschäftigten sich in verschiedenen Abhandlungen mit wissenschaftlicher Methodik [18]. Benjamin Franklin (1743) definierte nützliche Wissenschaft als:

*“...an objective process of studying important problems by comparing multiple hypotheses using experiments (designed, quasi, or natural). The process uses cumulative scientific knowledge and systematic measurement to obtain valid and reliable data, valid and simple methods for analysis, logical deduction that does not go beyond the evidence, tests of predictive validity using out-of-sample data, and disclosure of all information needed for replication.” [18]*

und führte damit als Prämisse die Offenlegung aller notwendigen Informationen zur Reproduktion der beschriebenen Experimente ein.

Damit wissenschaftliche Studien von der Gesellschaft akzeptiert werden, ist deren Reproduzierbarkeit ein Schlüsselkriterium. Verwendete Materialien, Methoden und technische Daten müssen ausreichend transparent sein, um Messungen und Experimente wiederholen zu können [23], wobei der Erfolg und die Glaubwürdigkeit von Studien stark von der Bereitschaft der Forscher abhängt, ihre Ideen und Ergebnisse anderen Forscher für unabhängige Reproduktionen in allen Details zu enthüllen [21].

Übertragen auf computerunterstützt erstellte Studien ist Reproduzierbarkeit als letztendliches Kriterium zu sehen, nach dem wissenschaftliche Forschung beurteilt wird [22].

### 2.2.2. Problematik

Die Wiederholbarkeit von Experimenten anderer Wissenschaftler sowie von eigenen Messungen stellt eine große Schwierigkeit dar. Aus einer Untersuchung von 1576 befragten Wissenschaftlern aus verschiedenen Sparten ging hervor, dass mehr als 70% am Versuch ein

fremdes Experiment zu reproduzieren, und mehr als 50 % beim Versuch eigene Experimente zu wiederholen, gescheitert sind [19].

Allgemein sind häufige Gründe für schlecht reproduzierbare Studien Zeitdruck bis zur Publikation, selektive Darstellung von Ergebnissen sowie inadäquate Anzahl wiederholter Messungen oder schwache statistische Signifikanz und seltener die Verwendung spezieller Methoden, welche eine Reproduktion erschweren [19].

Im Hinblick auf Reproduzierbarkeit von Studien, welche computerunterstützte Messungen oder Experimente enthalten, sind die drei folgenden Kriterien aus [16] zu betrachten:

- Transparenz - Definiert, ob Programmcode und Messdaten komplett - inklusive des Quellcodes aller Programmteile - oder nur teilweise (z.B. nur Ausführungsdatei und Programmteile) zur Verfügung gestellt werden.
- Übertragbarkeit - Definiert die Möglichkeiten der Übertragbarkeit der Messungen/des Programms auf unterschiedliche Systeme im Hinblick auf Prozessorarchitektur, Betriebssystem sowie ausreichende Dokumentation von verwendeten Programmen.
- Umfang - Definiert die Vollständigkeit der durchgeführten Arbeitsschritte des experimentellen Prozesses - z.B. komplette Dokumentation aller Arbeitsschritte von den Rohdaten bis zum fertigen Dokument inklusive aller Zwischenschritte des Workflows.

Daraus ergeben sich computerspezifische Gründe schlecht reproduzierbarer Studien wie fehlender, Messdaten erzeugender Quellcode oder Programmierfehler, welche für lange Zeit nicht oder nie korrigiert werden [16].

Eine weitere Problematik ergibt sich bei computerunterstützten Studien bei der Archivierung der Messdaten, Programmteile, Ergebnisse und Dokumente. Für die Bereitstellung und Archivierung können drei Ebenen definiert werden:

Die physische Ebene enthält die Daten in Form eines kodierte Informationsflusses - einer Abfolge von Bits - und wird hauptsächlich zur Archivierung verwendet.

Die logische Ebene stellt Daten in Form von interpretierbaren Formatspezifikationen dar.

Und die konzeptionelle Ebene, welche die Daten für den Mensch durch ausführbare Dateien interpretierbar darstellt. Für transparente, übertragbare Daten und Quellcode in vollem Umfang müssen alle drei Ebenen gesichert werden [17].

Intuitiv sollte die Sicherstellung der Reproduzierbarkeit bei computerunterstützten Systemen einfacher sein, da diese nicht den natürlichen biologischen Unsicherheiten unterliegen [1]. Zusätzlich besteht bei Programmen meist die Möglichkeit alle durchgeführten Aktionen automatisch detailliert zu dokumentieren [22]. Sind keine adäquaten Werkzeuge zur automatischen Dokumentation vorhanden oder sind diese ungenügend dokumentiert, ist es für Anwender aber oft mühselig Werkzeuge zu finden, deren Funktion zu verstehen und anzupassen. In letzter Konsequenz müssten Werkzeuge zur vollständigen automatischen Dokumentation neu erstellt werden.

Außerdem ist es schwierig die Gesamtheit aller, zur Reproduzierbarkeit benötigten Daten - wie Entwicklungsumgebung, Quellcodestücke, technische Spezifikationen, Hard- und Softwareparameter sowie Softskills und Zusatzinformationen zur Anwendung des Programms - zu erfassen [16].

### 2.2.3. Methoden und Aufgaben

Methoden zur Erzeugung reproduzierbarer Studien werden zur Genüge erstellt und publiziert, wobei ganz allgemein zwei Vorgehensweisen unterschieden werden können. Einerseits kann ein Plan für Reproduzierbarkeit erstellt werden, bevor mit der eigentlichen Arbeit begonnen wird und andererseits kann nach getaner Forschungsarbeit durch nachträgliche Dokumentation des Messaufbaus und Versuchsdurchführung die Reproduzierbarkeit sichergestellt werden [16]. Oft wird bei einem gut reproduzierbaren Projekt eine Mischung der beiden Methoden resultieren.

Die zugrundeliegenden Aufgabenbereiche werden in [16] grob eingeteilt in:

- Erstellung einer detaillierten Beschreibung des Experiments inklusive erzeugenden Quellcode.
- Zusammenstellung eines Programmpakets aller verwendeten Komponenten des Experiments.
- Verknüpfung der veröffentlichten Studien mit den Messdaten und Experimenten.
- Reproduktion und Verifikation der Messergebnisse.

Viele Publikationen bieten Checklisten der Vorgehensweise, die Reproduzierbarkeit von Studien sicherzustellen, an. In [18] und [25] sind derartige Checklisten ersichtlich, wobei die „Zehn einfachen Regeln für reproduzierbare computerunterstützte Forschung“ aus [25] überblicksmäßig anschließend angeführt werden:

1. Verfolge für jedes Ergebnis dessen Entstehungsweg
2. Vermeide manuelle Manipulationen an Daten
3. Archiviere exakte Versionen aller internen und externen verwendeten Programme
4. Behalte die Übersicht über die Versionen von individuell angefertigten Skripten
5. Dokumentiere alle Zwischenergebnisse, wenn möglich in standardisierten Formaten
6. Für Analysen, welche Zufallsgeneratoren beinhalten, speichere Initiatoren (random seeds)
7. Speichere immer Rohdaten, die den Analysen und Diagrammen zugrunde liegen
8. Generiere hierarchische Analyseergebnisse, welche Schichten unterschiedlicher Genauigkeit aufweisen
9. Verknüpfe Textpassagen mit entsprechenden Messergebnissen
10. Veröffentliche Quellcode, ausführbare Dateien und Messergebnisse

In [17] und [20] werden Aufgaben, Kriterien und Möglichkeiten der Langzeitarchivierung, Bereitstellung und Publikation von Studien zur Erhöhung der Reproduzierbarkeit mittels Forschungsdaten-Repositoryn, wie RADAR(**R**esearch **D**ata **R**epositorium), diskutiert. Weitere Herangehensweisen und detailliertere Beschreibungen von Grundprinzipien wissenschaftlicher Integrität der OeAWI (Österreichische **A**gentur für **w**issenschaftliche **I**ntegrität) sind in [24] zu finden.

### 2.2.4. Ableitungen

Schlussfolgerungen für diese Arbeit sind in zwei verschiedenen Aspekten zu treffen:

Einerseits müssen die Messergebnisse bei Verwendung des Programms, sowie deren Entstehung nachvollziehbar und reproduzierbar sein, damit die Reproduzierbarkeit der - aus den Messdaten resultierenden - Studien sichergestellt ist.

Andererseits muss diese Arbeit grundlegend reproduzierbar sein, um das Verständnis der Funktionsweise des Programms zu vermitteln und um Modifikationen oder ein neues Setup des Messaufbaus durchführen zu können.

Wie bereits erwähnt wird die Methodik zur Sicherstellung der Reproduzierbarkeit oftmals teils vorher geplant und teils im Nachhinein erfolgen. Auch in dieser Arbeit werden um die Reproduzierbarkeit zu erhöhen, automatische Dokumentations-Routinen geplant und implementiert. Die Erstellung des Programms und der betreffenden Routinen wird klarerweise im Nachhinein dokumentiert.

Bei **Verwendung des Programms** zur Erzeugung von Daten für weitere Studienzwecke werden zur Sicherstellung der Nachvollziehbarkeit und Reproduzierbarkeit folgende Komponenten geplant:

- Während der Messung, Analyse und Auswertung werden von den jeweiligen Routinen automatisch Funktionsparameter sowie durchgeführte Arbeitsschritte des Workflows dokumentiert.
  - Dafür wird das Konzept der Referenzierung aller Arbeitsschritte entwickelt.
  - Für anwendungsspezifische Parametereinstellungen müssen Parameter variabel sein. Diese Funktionsparameter werden in entsprechenden Logfiles dokumentiert.
- Manuelle Manipulationen von Daten während der Verwendung müssen weitestgehend unterdrückt werden.
  - Nicht vermeidbare Manipulation muss ausreichend automatisch dokumentiert werden.
- Das Programm wird als White-Box zur Verfügung gestellt, damit detailliertere Beschreibungen der Funktionsweise in resultierenden Papers möglich sind.
- Rohdaten sowie Zwischenergebnisse der Auswertung werden als Ausgabedateien im standardisiertem CSV-Format bereitgestellt.
- Arbeitsschritte einzelner Mess-, Analyse- und Auswertungssequenzen können individuell variieren. Diese werden detailliert dokumentiert um „Deep Caption“ von Diagrammen/Tabellen zu ermöglichen. „Deep Caption“ bedeutet, dass darstellenden Ergebnissen im Paper der Workflow zur Entstehung der Ergebnisse - wie Experimentaufbau - hinterlegt wird [17].

Da die Messergebnisse für vielseitige Anwendungen verwendet werden können, muss für resultierende Studien individuell entschieden werden, welche Daten und Methoden zur Sicherstellung der Reproduzierbarkeit benötigt werden. Hier ist anzumerken, dass Reproduzierbarkeit allein selbstverständlich nicht die Korrektheit und Qualität von resultierenden Studien garantiert [22].

Zur Sicherstellung der Nachvollziehbarkeit **dieser Arbeit** sind folgende Komponenten vorgesehen.

- Die oben angeführten Kriterien zur Reproduzierbarkeit von Routinen sind folgendermaßen definiert:

#### Transparenz - komplett

- Alle Programmteile (Klassen, Skripten) und Stufen (Quellcode, EXE-Datei) werden zur Verfügung gestellt.

#### Übertragbarkeit - eingeschränkt auf ähnliche Messsysteme

- Das Programm soll auf gleiche und ähnliche Systeme ohne bzw. mit geringfügigen Änderungen übertragbar sein. Dazu wird der aktuelle Messaufbau detailliert beschrieben und Setup sowie Modifikationsmöglichkeiten des Programms erläutert.

#### Umfang - vollständig

- Alle Arbeitsschritte von der physischen Probe bis zu den Ausgabedateien werden vollständig erklärt und transparent zur Verfügung gestellt.

- Für das Setup des Programms wird ein Paket aller Komponenten des Programms, ersichtlich in Abbildung 6, inklusive

- Des Ordner mit allen benötigten Daten zur Verwendung - „CTLFILES“
  - Nukliddateien für Auswertungen - „nuklide\_mittel.csv“, „nuklide\_lang.csv“
  - Parametertabelle für Spezialanalysen - „peaks.txt“
  - Standardvorlage für Standardanalysen - „a\_heinz.tpl“
  - Spezialvorlage für Spezialanalysen - „sanal.tpl“
- dem Quellcode des Hauptprogramms - „DUAL\_LFC“
- dem Quellcode der Probenwechsler-Routine „SOURCE\_PW“
- der EXE-Datei des Probenwechslers - „DLFC\_ADV.exe“
- der EXE-Datei des Hauptprogrammes - „DUAL\_LFC“
- der Bibliothek für Probenwechsler-Subroutine - „inpout32.dll“
- dieser Arbeit inkl. Bedienungsanleitung - „Manual.pdf“

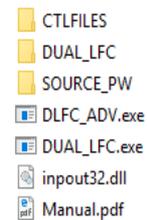


Abb.6:  
Inhalt des Archivs

- Das erstellte Komplettpaket wird zur Archivierung gepackt und an beiden Messsystemen im Ordner C:\Program Files\AutoMeasure\_DLFC\ sowie auf einer CD gesichert.
- Zum Verständnis der Funktionsweise und der Nachvollziehbarkeit der Erstellung des Programms, wird
  - jede vorhandene Subroutine aller Klassen angeführt und deren Funktion beschrieben
  - die Entwicklungsumgebung dokumentiert.
- Zur Verdeutlichung der Anwendungsmöglichkeiten werden eine ausführliche und eine kurze Bedienungsanleitung erstellt.

Alle Komponenten zur Erhöhung der Reproduzierbarkeit bei der Verwendung des Programms sowie in dieser Arbeit sollten weitest möglich realisiert werden.

## 3. NEUORGANISATION DES MESSAUFBAUS

### 3.1. VORBEREITUNGEN

Das Messsystem zur INAA beinhaltet zwei Gammastrahlendetektoren „Russe“ und „dicker Fritz“. Beide Detektoren sind HPGe und ausgestattet mit Zählratenkorrektursystemen der Korrekturmethode LFC (Kapitel 2.1.2.3.). An die Detektoren sind zwei Messrechner, PC1 und PC2, angeschlossen. An den Detektor „Russe“ ist zusätzlich ein Probenwechsler gekoppelt, welcher mit PC1 gesteuert werden kann. Der Detektor „Russe“, PC1 und der Probenwechsler bilden ein Messsystem - folgend als Messsystem-Russe bezeichnet - und der Detektor „dicker Fritz“ bildet mit dem PC2 ein Messsystem, bezeichnet als Messsystem-Fritz.

#### 3.1.1. IST-Zustand

**PC1** - Betriebssystem: Windows 98

Dieser PC ist stark veraltet und fehleranfällig. Durch fehlende Updates und große Technologiespanne, kann am PC1 kein USB-Stick verwendet werden, da der PC ansonsten abstürzt. Dadurch ist die Datenübertragung der Messergebnisse stark erschwert via Diskette oder CD-ROM durchzuführen. Zusätzlich benötigt der PC für Neustarts teilweise mehr als eine Stunde um wieder funktionsfähig zu sein.

**PC2** - Betriebssystem: Windows 7

Dieser Mess-PC wurde 2017 bereits erneuert. Beim Austausch des veralteten PC2 zu dem aktuellen Mess-PC2 wurde festgestellt, dass die bis dato verwendete Software zur Messung von Probenreihen „DLFC“ und zur Analyse von Messspektren „DLFCANAL“ am PC2 mit dem Betriebssystem Windows 7 nicht mehr funktionsfähig war. Ein Problem war die Ansteuerung des Probenwechslers mit der vorhandenen Routine. Diese führte unter Windows 7 zu einer Fehlermeldung und zum Absturz des Programms.

#### 3.1.2. SOLL-Zustand

PC1 und PC2 sollen vollkommen identische Rechner mit gleichen Spezifikationen sein. Beide sollen die gleiche Software und die neuesten Updates bzw. eine aktuelle aber vor allem gleiche Version des .NET Framework aufweisen, um hinzukommende Software leichter implementieren zu können. Dies dient einerseits zur Erleichterung von regelmäßigen Updates beider Messcomputer, andererseits als Vorbereitung für einen zusätzlichen Probenwechsler, welcher an den „Russen“ gekoppelt werden soll, damit auch mit diesem Detektor die Möglichkeit besteht Serienmessungen durchzuführen.

#### 3.1.3. Problematik

Das Messsystem-Russe wird aktuell für Serienmessungen verwendet. Dementsprechend muss vor einer Erneuerung dieser Komponenten sichergestellt werden, dass Serienmessungen problemlos durchgeführt werden können, wenn der Messrechner in einer Pause zwischen zwei Serienmessungen, ausgetauscht wird. Diese Einschränkung betrifft nur den Teil

der Messung, da es vorübergehend Möglichkeiten gibt die Analyse der Spektren anderweitig durchzuführen. Die Auswertung wird in dieser Phase mit der bisherigen Prozedur durchgeführt.

## 3.2. DURCHFÜHRUNG

Zuerst wird die Software am Messsystem-Fritz auf den neuesten Stand gebracht und die Ansteuerung des Probenwechslers unter Windows 7 ermöglicht. Danach wird eine vorläufige Version des ersten Programmteils - Messung - erstellt und umfangreich getestet. Anschließend wird der neue, für das Messsystem-Russe vorgesehene PC so weit wie möglich vorbereitet, um in möglichst kurzer Zeit einen Austausch der Rechner durchführen zu können. Dies ermöglicht eine Testphase auf dem Messsystem-Fritz um die Funktionstüchtigkeit sicherzustellen.

### 3.2.1. Updates

Auf dem PC2 - ein IBM-Rechner mit den Spezifikationen

**Prozessor:** Intel® Core™ i5-3550S CPU@3.00 GHz  
**Arbeitsspeicher:** 4.0 GB  
**Systemtyp:** 32-Bit  
**Computername:** radiochemie\_dFritz

wurde das Update-Pack

„WinFuture\_7SP1\_x86\_UpdatePack\_2.81\_November\_2017-Vollversion“ für Windows 7 - Servicepack 1 installiert.

Da die Installation der neuesten Version des .NET Frameworks verweigert wurde, musste das .NET Framework stufenweise mit folgenden Versionen erneuert werden:

1. Version 3.5: dotnetfx35\_full\_offline
2. Version 4.5.2: dotNetFramework\_NDP452-KB2901907-x86-x64-AILOS-ENU
3. Version 4.6.2: NDP462-KB3151800-x86-x64-AILOS-NEU

### 3.2.2. Probenwechsler

Der Probenwechsler ist über eine parallele Schnittstelle mit PC1 verbunden. PC1 verfügt über zwei parallele Schnittstellen - LPT2 und LPT3 (line printing terminal).

Für LPT2 ist der Adressbereich D0A0 - D0A7 und D0B0 - D0B7, welche den Adressnummern 53408 - 53415 und 53424 - 53431 entsprechen, reserviert.

Für LPT3 ist der Adressbereich D040 - D047 und D050 - D057, welche den Adressnummern 53312 - 53319 und 53328 - 53335 entsprechen, reserviert.

Eine Adressnummer entspricht einem Byte - Binär: „00000000“ bis „11111111“ gleich Dezimal: „0“ bis „255“. Durch setzen eines Wertes von „255“ an dieser Adresse wird das gesamte

Byte „1“ = „WAHR“ gesetzt. „1“ entspricht einer Steuerspannung von 5V. Wird ein Byte WAHR gesetzt, wird der adressierte Pin mit einer Spannung von 5V versorgt.

Der Probenwechsler ist mit einem der beiden LPT-Ports verbunden und kann durch Setzen des Kontrollpins auf WAHR der parallelen Schnittstelle aktiviert und deaktiviert werden. Da der verwendete LPT-Port und der vom Probenwechsler als Kontrollpin verwendete Pin unbekannt sind, müssen die angeführten Adressbereiche auf Erfolg getestet werden.

Es steht ein Programm „PORTDATA\_InpOut\_32.exe“ zur Verfügung, welche durch Eingabe der Adressen in hexadezimalform die adressierten Bytes setzt. Für diese ausführbare Datei, steht leider kein Quellcode zur Verfügung. Dieses Programm wird dazu verwendet den Kontrollpin des Probenwechslers zu bestimmen.

Durch setzen des Pins mit der **Adresse 53411 bzw. D0A3** auf WAHR und FALSCH lässt sich der Probenwechsler steuern.

Nun muss eine Routine erstellt werden, welche während des Messvorganges aufgerufen werden kann, um mehrere Proben nacheinander zu messen.

Es ist bereits ein nicht funktionstüchtiges Programm - porttest32, codiert in Pascal - vorhanden. In Anlehnung an diese Routine wird unter Verwendung der freien Entwicklungsumgebung Lazarus für die Programmiersprachen Pascal und Object Pascal eine Routine - „DLFC\_ADV“ - erstellt. Der komplette Code befindet sich im Anhang B. Vereinfacht enthält die Routine folgende Komponenten:

1. die Definition des anzusprechenden Pins
2. setzen des Pins auf WAHR
3. abwarten eines Delays von 3 Sekunden
4. setzen des Pins auf FALSCH

Um den Wert eines adressierten Pins im Speicher auszulesen und zu setzen, werden Funktionen inp32 und out32 verwendet, welche in einer Klasse „inpout“ deklariert sind. Diese Funktionen stellen den CPU-Typ des Mess-PC's fest und rufen die entsprechende Bibliothek auf, mittels derer der Pin gesetzt wird. Da beide Messsysteme Prozessortypen von 32 Bit aufweisen, wird nur die Bibliothek inpout32.dll benötigt. Die Klasse inpout wird aus dem bestehenden Programm übernommen, da diese funktionstüchtig und somit keine Neuprogrammierung notwendig ist. Zur besseren Kontrolle sind mehrere Ausgabemöglichkeiten implementiert. In Abbildung 7 ist eine Standarddurchführung mit Ausgabe ersichtlich.

Die Routine DLFC\_ADV.exe wird fixiert am Pfad "C:\Program Files\AutoMeasure\_DLFC\DLFC\_ADV.exe" gespeichert. Im gleichen Ordner AutoMeasure\_DLFC muss sich die Bibliothek inpout32.dll befinden, um aufgerufen werden zu können. Das Programm DLFC wird ebenfalls an diesem Ort abgelegt.

Durch Ausführen von DLFC\_ADV wird der Probenwechsler gestartet. Dies bedeutet, dass eine sich im Detektor befindliche Probe durch Druckluft aus dem Detektor befördert wird. Anschließend wird die nächste Probe ebenfalls durch Druckluft im Detektor platziert. Einige Parameter wie die Dauer der Extraktion- und Implantationsphase, sowie der zeitliche Abstand der beiden Phasen können manuell am Probenwechsler variiert werden.

```
Port: 53411
ON/OFF: 0

PIN gleich 1 setzen.
ON/OFF: 1

PIN gleich 0 setzen.
ON/OFF: 0
```

Abb.7: Funktionsablauf -DLFC-ADV; Screenshot des Verfassers

### 3.2.3. Neuaufbau Messsystem-Russe

Nach der Erstellung des vorläufigen ersten Programmteils „Messung“ welcher in Kapitel 4.3. näher beschrieben wird, folgt der Austausch des alten PC1 mit dem neuen Messrechner.

Der neue PC1 ein IBM-Rechner mit den Spezifikationen

**Prozessor:** Intel® Core™ i5-3550S CPU@3.00 GHz  
**Arbeitsspeicher:** 4.0 GB  
**Systemtyp:** 32-Bit  
**Computername:** radiochemie\_Russe

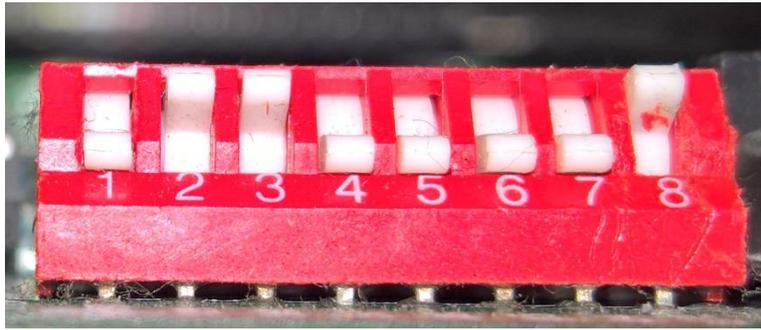
wird mit Windows 7 eingerichtet und ServicePack 1 versehen. Anschließend werden ebenfalls die Updates „WinFuture\_7SP1\_x86\_UpdatePack\_2.81\_November\_2017-Vollversion“ sowie die Versionen des .NET Frameworks 3.5, 4.5.2. und 4.6.2. installiert.

Für die Verwendung der Software Genie 2000 wird eine ISA-Messkarte (**I**ndustry **S**tandard **A**rchitecture) der Marke AccuSpec benötigt. Diese wird aus dem alten PC1 ausgebaut und im neuen PC1 installiert. Hierbei ist auf die benötigte Jumperstellung zu achten, welche mit der auszuwählenden In- und Output Adressierung übereinstimmen muss.

Verwendet wird die Messkarte vom Typ AccuSpec B, welche die in Abbildung 8 ersichtliche Jumperstellung aufweist.

Anschließend kann auf dem neuen PC1 die Messsoftware Genie 2000 Basisspektroskopie inklusive Sentinel Software installiert werden. Im Zuge dieser Installation wird automatisch der Vielkanalanalysator-Eingangseditor (VKA Editor) installiert. Ist die Installation abgeschlossen, muss im VKA Editor die Messkarte eingerichtet werden. Zuerst muss unter *Bearbeiten - Hinzufügen VKA - Steckkarten VKA's - AccuSpec B* die Messkarte zugefügt werden.

CLOSED - OPEN - OPEN - CLOSED - CLOSED - CLOSED - CLOSED



**Abb.8:** Stellung der Jumper an der Messkarte - AccuSpec B; Screenshot des Verfassers

In *Geräte - VKA* sind folgende Einstellungen zu konfigurieren:

- 16K ist auszuwählen
- Anzahl der ACD's: „1“
- I/O Adresse: „0300“
- Hex: „D000“

Die I/O Adresse ist definiert durch die o.a. Jumperstellung, welche der Tabelle 3 aus dem Genie Manual [30] entnommen wurde.

Board No.:	1	2	3	4
I/O Addr Switch:	300	310	320	330
A10	CLOSED	CLOSED	CLOSED	CLOSED
A9	OPEN	OPEN	OPEN	OPEN
A8	OPEN	OPEN	OPEN	OPEN
A7	CLOSED	CLOSED	CLOSED	CLOSED
A6	CLOSED	CLOSED	CLOSED	CLOSED
A5	CLOSED	CLOSED	OPEN	OPEN
A4	CLOSED	OPEN	CLOSED	OPEN

**Tabelle 3:** AccuSpec B - Adresseinstellungen

Die Bezeichnungen A10 - A4 entsprechen den Jumpernummierungen 1-7. Der Jumper Nr. 8 muss „OPEN“ sein, damit die übrigen Jumperstellungen berücksichtigt werden.

Unter *Einstellungen - Eingang* müssen

- Eingangsname: „DLFC“
- Detektortyp: „Gamma- od. Röntgendetektor - Ge“
- Eingangsgröße: „16384“ Kanäle

ausgewählt werden. Hier ist der Eingangsname „DLFC“ nötig, damit der Detektor in Genie2000 angezeigt wird. Abschließend muss der Detektor mit *Datenbank - Laden in - DLFC* geladen werden. Beim Mess-PC 2 werden diese Einstellungen im Zuge einer Computernamenänderung ebenfalls neu eingerichtet. Die einzige Änderung zur obigen Konfiguration betrifft die Jumperstellung der AccuSpec B Karte im PC2, welche zu I/O Adresse: „0320“ führt. Nun kann mit der Software Genie 2000 - Gammamessung und -analyse ein Gammastrahlenspektrum aufgenommen werden.

Damit gemessene Peaks annähernd korrekten Energielinien, also ihren jeweiligen Energien, zugewiesen werden können, muss der Detektor grob kalibriert werden. Dazu werden Kalibrierproben -  $^{241}\text{Am}$ ,  $^{137}\text{Cs}$  und  $^{60}\text{Co}$  - im Detektor platziert und eine Minute gemessen. Eine derart kurze Messzeit reicht, damit die Ausprägungen der Energielinien der Proben stark genug sind, um die Peaks den Linien zuzuweisen und somit eine grobe Kalibrierung durchzuführen.

Es werden in *Kalibrierung - Energie & Peakform - mit Nuklidliste* die angeführten Isotope von Americium, Cäsium und Kobalt ausgewählt und nacheinander alle Energielinien den Peaks im Spektrum zugeordnet, indem der Cursor an der Peakspitze positioniert und der Button „Cursor“ betätigt wird. Dies erzeugt eine lineare Beziehung zwischen den Kanälen des Spektrums und den Energieniveaus der Peaks. Zur Kontrolle sollte mit dem Button „Zeigen“ die Linearität der Kalibrierung überprüft werden. Kalibrierpunkte, welche von der Geraden weit entfernt sind, sollten gelöscht werden. Die abgeschlossene Kalibrierung wird direkt am Detektor gespeichert.

Um ein funktionsfähiges Messsystem zu gewährleisten, werden umfangreiche Tests durchgeführt und einige Sicherheitsabfragen implementiert. Die vollständige Beschreibung der Fehlerbehandlung ist ein Teilbereich des nachfolgenden Kapitels.

## 4. ERSTELLUNG DES PROGRAMMS - DUAL\_LFC.EXE

Dieses Programm soll die Arbeitsschritte Messung, Analyse und Auswertung zur INAA möglichst automatisieren. Der zu optimierende Workflow beginnt nach den Schritten der Probenvorbereitung und Aktivierung der Proben, welche in Kapitel 2.1.2. kurz beschrieben sind.

Die Messung umfasst die Aufnahme des Gammaskpektrums der Proben, sowie die Teilung des kompletten 16K Spektrums(raw) in korrigiertes (LFC - **L**oss **F**ree **C**ounting) und unkorrigiertes (UNC - **U**ncorrected) Spektrum.

Die Analyse behandelt vorrangig die speziell für die Anwendung angepasste Peak-Suche und den Peak-Fit, inklusive der Integration der Peak-Fläche. Zusätzlich beinhaltet der Abschnitt der Analyse die Möglichkeiten der nachträglichen Kalibrierung von Spektren, der Analyse von Spektren mit beliebig auswählbaren Analysesequenzen und einer Funktion „Interaktiver-Peak-Fit“ zur Kontrolle und Verbesserung von gefitteten Peakflächen.

Im Teil „Auswertung“ werden aus den erhaltenen Peakflächen und einigen Standards, welche Teil der Probenreihe sind, die Konzentrationen [ $\mu\text{g/g}$ ] - wieviel Mikrogramm des Elements pro Gramm Probeneinwaage in der Probe enthalten sind - der Elemente berechnet.

Die R-Werte der einzelnen Standards erreichen für problematische Radionuklide nicht immer die gleiche Qualität. Daher müssen die Standard-Werte überprüft werden. Dieser Teil der Auswertung kann nicht vollständig mit höchstmöglicher Ergebnisqualität automatisiert werden und sollte daher meist manuell durchgeführt werden.

Sehr vereinfacht können die drei Schritte zusammengefasst werden:

- Im ersten Schritt werden Spektren aufgenommen und Ausgabedateien - Spektren(.cnf): *dateinamen\_LFC.cnf*, *dateinamen\_UNC.cnf* - für jede Probe erstellt.
- Anschließend werden im zweiten Schritt diese Spektren analysiert und Reportdateien(.rpt): *dateinamen\_LFC.rpt*, *dateinamen\_UNC.rpt* erzeugt.
- Im dritten abschließenden Schritt werden die Daten aus der Reportdatei ausgelesen und ausgewertet und Ausgabedateien - Ergebnis- und Dokumentationstabellen(.csv): *PRN\_logfile.csv*, *PRN\_messwerte.csv*, *PRN\_standards\_init.csv*, *PRN\_standards\_fin.csv* und *PRN\_auswertung.csv* - zur weiteren Verwendung erstellt. PRN steht für Probenreihenname und spezifiziert die Dateinamen der Ausgabedateien.

In jeder Klasse werden Funktionen, welche dazu dienen fehlerhafte Benutzereingaben, leicht verwechselbare Einstellungen, Programmabstürze und Endlosschleifen im Programm zu vermeiden, extra in den Kapiteln „Fehlerbehandlung“ ausgeführt. Diese Kapitel beziehen sich demnach immer nur auf die Behandlung von Fehlern, welche aus dem Programmverlauf und dessen Benutzung resultieren und nicht auf die Behandlung von Messfehlern.

## 4.1. VORBEREITUNGEN

### 4.1.1. Genie 2000

Bei den Teilen „Messung“ und „Analyse“ ist die Implementierung von Prozeduren und Funktionen der Klassen des Programms Genie2000 notwendig. Dementsprechend muss vor Programmierbeginn die Funktionsweise des Programms Genie2000 studiert werden. Dazu werden hauptsächlich die Handbuch-Komponenten

- Genie 2000 Read Me First
- Genie 2000 Operations Manual [26]
- S506 Interactive Peak Fit User's Manual [27]
- Genie 2000 Customization Tools Manual [28]
- S561 Batch Tools Support [29]
- Historical Canberra MCAs [30]
- Genie 2000 Tutorials Manual[31]

verwendet, welche in der Installation von Genie2000 inkludiert sind.

Grundsätzlich wurden zwei geeignete Herangehensweisen an die Verwendung von Genie-Routinen extrahiert:

1. Aufruf von BATCH Prozeduren in der Kommandozeile
2. Einbinden von Bibliotheken - COM DLL's (**C**omponent-**O**bject-**M**odel **D**ynamic-**L**ink-**L**ibrary)

#### 1. BATCH Prozeduren

Die wichtigsten Aufgaben von Genie sind in Prozeduren implementiert, welche durch Kommandozeilen-Befehle betreffende Spektroskopie-spezifische Programmmodule aufrufen. Die Prozeduren rufen kompilierte ausführbare EXE-Dateien auf, deren Funktionsweise mit Merkmalen spezifiziert werden können.

z.B.: „STARTMCA det:DLFC /livepreset=10000

*Vorteile:*

- |                       |   |
|-----------------------|---|
| <b>Einfachheit</b>    | - Nur eine kleine Anzahl an Einschränkungsmerkmalen ist vorhanden.            |
| <b>Stabilität</b>     | - Die aufgerufenen Prozeduren sind bereits kompilierte EXE-Dateien.           |
| <b>Sicherheit</b>     | - Die Fehlerbehandlung wird auch in aufgerufenen Modulen durchgeführt.        |
| <b>Kompatibilität</b> | - Spätere Versionen von Genie stellen dieselben EXE-Prozeduren zur Verfügung. |

*Nachteile:*

- Geschwindigkeit** - Der Aufruf der Prozeduren in der Kommandozeile ist langsam.
- Komplexität** - Für komplexere Anwendungen fehlen detailliertere Einstellungsmöglichkeiten.

## 2. COM DLL Funktionen

Das Programm Genie2000 ist in Klassen aufgebaut, deren Funktionen durch die Einbindung von verwalteten DLL zugänglich sind. Dazu müssen in Visual Studio Verweise auf die entsprechenden Bibliotheken hinzugefügt werden. Dadurch können die Funktionen und deren Syntax sowie Parameter der eingebundenen Klasse im Objektkatalog eingesehen werden. Es existiert ein Handbuch „S560 Genie 2000 Programming Library“, welches in dem vorhandenen Canberra Setup nicht inkludiert ist und dadurch leider nicht zur Verfügung steht.

Manche dieser älteren Bibliotheken erzeugen bei Verwendung in neueren .NET Frameworks Fehler. Die COM DLLs müssen im Projekt eingebunden werden und somit kompiliert werden um ausführbare Dateien zu erhalten. Verwendet werden folgende Bibliotheken:

- CanberraDataAccessLib [DataAcc.dll]
- CanberraSequenceAnalyzerLib [SeqAnlyz.dll]
- CanberraReporterLib [rptview.dll]

*Vorteile:*

- Diversität** - Die Funktionen bieten viele Parameter und Auswahlmöglichkeiten zur Spezifikation der Durchführung.
- Geschwindigkeit** - Der mehrmalige Aufruf von eingebundenen Funktionen wird während der Kompilierung optimiert und ist somit sehr viel schneller.

*Nachteile:*

- Stabilität** - Die Komplexität der Klassen und die große Anzahl verschiedener Parameter, können in Ausnahmefällen durch nicht oder falsch indizierte Parameter Fehler erzeugen.
- Komplexität** - Die Auswirkungen durch Verschachtelungen von mehreren Funktionen müssen bedacht werden.

Die Möglichkeit der Eingliederung von Genie Prozeduren führt zu umfassenden Recherchen in den Handbüchern, um fundierte Entscheidungen treffen zu können, welche Routinen in welcher Form eingebunden werden sollen.

### 4.1.2. Visual Basic(VB) - .NET Framework

Die Auswahl der Programmiersprache gründete sich auf folgende Kriterien:

- Eine grafische Benutzeroberfläche (GUI - graphical user interface) wird erwünscht, um Anwendereingaben zu erleichtern.
- Für Messung und Analyse sind alte Routinen vorhanden, welche ebenfalls in Visual Basic verfasst sind - diese können als Hilfestellung verwendet werden.
- Es müssen einige sehr unterschiedliche Komponenten - Hardwaresteuerung bei Messung und Probenwechsler, Prozesseinbindung externer Software, Einbindung externer Dateien - in diesem Programm vereinigt werden.

Als Programmierumgebung wird Microsoft Visual Studio Enterprise 2017 - Version 15.4.4 verwendet. In den Projekteigenschaften müssen der Anwendungsname, der Stammordner, die Verweise der einzubindenden COM-DLL's, das Zielframework und die Ziel-CPU definiert werden, wobei auf den 32 Bit Prozessortyp der Mess-PC's zu achten ist.

### **4.1.3. Reproduzierbarkeit**

Aufgrund der Gliederung des Projektes werden vier Hauptklassen

- DLFC\_index
- Messung
- Analyse
- Auswertung

definiert, deren Erstellung im folgenden Kapitel einzeln erörtert wird.

Durch alle Klassen zieht sich die Schwierigkeit der Reproduzierbarkeit des INAA-Prozesses und der damit einhergehenden wissenschaftlichen Aussagekraft der gewonnenen Daten.

Wichtigste Direktive ist die Nachvollziehbarkeit, welche Arbeitsschritte - seien es automatisierte oder manuelle - an Daten durchgeführt wurden, sowie deren Dokumentation bzw. Sicherung.

In den folgenden Kapiteln werden hinsichtlich der Reproduzierbarkeit die Problematik der einzelnen Bereiche und die daraus resultierenden Maßnahmen herausgearbeitet.

## **4.2. KLASSE - DLFC\_INDEX.VB**

### **4.2.1. Problematik der Reproduzierbarkeit - Stammklasse**

Aktuell wird die Messung mit einem VB Programm - DLFC - durchgeführt, die daraus resultierenden Spektren mit einem zweiten VB Programm - DLFCANAL - analysiert. Anschließend werden die produzierten Report-Dateien mit einer weiteren VB-Routine in eine Excel-Datei - „NAA\_Auswertung.xls“ - übertragen. In dieser Excel-Datei finden die Berechnungen der Auswertung statt.

Die Probleme hinsichtlich Reproduzierbarkeit sind in diesem Abschnitt vor allem die Anzahl der Komponenten, die eine Nachvollziehbarkeit der Arbeitsschritte erschwert und der Bedarf an Benutzereingaben und Auswahlmöglichkeiten. Diese führen zu der Notwendigkeit einer ständigen Dokumentation der durchgeführten Arbeitsschritte und Benutzereingaben, wozu das Konzept der Referenzierung entwickelt wurde. Jeder Arbeitsschritt, welcher mit einem Spektrum durchgeführt ist, wird mit einem Kürzel in Beschreibungsparametern (CAM\_T\_SDESC1 und CAM\_T\_SDESC4) direkt in der Spektren-Datei gespeichert. Die Referenz wird auf zwei Parameter aufgeteilt, weil ein Parameter auf 64 Zeichen beschränkt ist. Folgend ist die Zusammensetzung der Referenzen der einzelnen Funktionen angegeben:

**Messung:**

aquisition - Datum [MMyy] - Spektrentyp [lfc/unc/dua]  
z.B.: Q0318lfc/

**Standardanalyse:**

Analyse - ausgew. Analysesequenz (.asf)  
z.B.: A5 SIGMA\_OR/

**Spezialanalyse:**

Special Analyze - ausgew. Peaktabelle (.txt) - Analysetyp [M/L]  
z.B.: SApeaksM/

**Kalibrieren:**

Calibrate - ausgew. Kalibrierdatei (.cal / .cnf)  
z.B.: CGc50/

**IPF:**

Interactive Peak Fit - Benutzereingabe nach IPF  
z.B.: ipf208,559/

Die vollständige Referenz wird bei der Analyse in der Reportdatei dokumentiert und könnte folgende Form haben.

ID: Q0318lfc/Cgut/ipf1077/A5 SIGMA\_OR/

Es ist darauf zu achten, dass nur Aktivitäten im Report-File dokumentiert sind, die vor der Analyse durchgeführt wurden. Somit wird sichergestellt, dass die verwendeten Daten im Report-File mit der Referenz übereinstimmen. Werden nach der Analyse weitere Arbeitsschritte durchgeführt, werden diese nur im Spektrum nicht aber in der Reportdatei dokumentiert. Sollen nachträgliche Veränderungen in die Auswertung miteinbezogen werden, muss eine weitere Analyse durchgeführt werden.

Bei der Auswertung werden alle Referenzen in einem Logfile gesammelt, um eine Übersicht der durchgeführten Maßnahmen zu erhalten.

## 4.2.2. Implementierung der Stammklasse

Diese Klasse ist die Stammklasse des Programms. Sie enthält die Ausgangsebene der grafischen Benutzeroberfläche (GUI), welche beim Ausführen des Programms aufgerufen wird und in Abbildung 9 dargestellt ist. Anhand der Abbildung kann bereits die Gliederung der Klassen erkannt werden. Die Gruppen „Aktuelle Messung“ und „Parameter“ sind in die Messklasse einzuordnen. Die Analyse-Klasse verfügt über die Gruppen „Kalibrieren“ und „Analyse“ und die Auswertung-Klasse besteht aus der Rahmengruppe „Auswertung“.

Die Bedienelemente in der linken unteren Ecke sind in der Stammklasse DLFC\_index implementiert.

Die Aufgaben der Stammklasse beinhalten die Handhabung der Benutzereingaben und deren Übergabe an Funktionsaufrufe aller generalisierten Funktionen, welche häufiger von den Unterklassen aufgerufen werden und den Großteil der Fehlerbehandlung. Die Funktion der automatischen aufeinanderfolgenden Durchführung aller Arbeitsschritte Messung - Analyse - Auswertung [AUTO] ist ebenfalls in dieser Klasse implementiert.

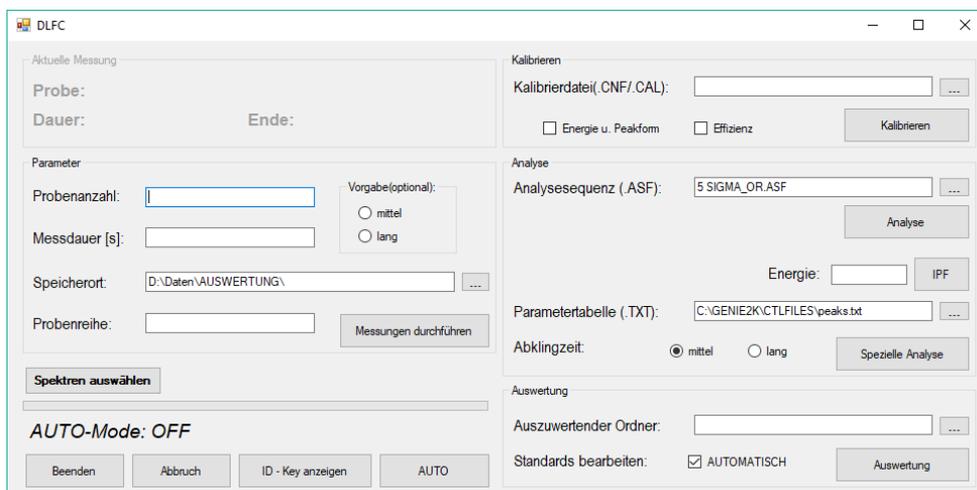


Abb.9: GUI des Programms DUAL\_LFC; Screenshot des Verfassers

### 4.2.2.1. Öffentliche Funktionen

Diese Funktionen können aus allen Klassen verwendet werden.

#### sleep\_fct(integer)

Ist eine Delay Funktion, welche die Ausführung des weiteren Codes für eine beliebige Anzahl an Sekunden stoppt. Vorrangig wurde diese Funktion für die Wartezeit während des Probenwechsels implementiert.

*Parameter:* Wartezeit in Sekunden; Datentyp Integer

#### update\_fct(string, integer)

Dient als Aufruf der Stammklasse, um während der Ausführung von Code anderer Klassen, Änderungen an der GUI zu bewerkstelligen. Vor allem Status-Anzeigen sowie die Fortschrittsleiste müssen regelmäßig aktualisiert werden.

*Parameter:* Status der Sequenz [anal/sanal/cal/ausw]; Datentyp String  
Modifizierung für Fortschrittsleiste; Datentyp Integer

#### 4.2.2.2. Allgemeine Funktionen

Bid\_key\_Click() bei Button-Click Event „ID-Key anzeigen“

Zeigt bei allen ausgewählten Spektren die zugehörigen Referenzen an

Bbeenden\_Click() bei Button-Click Event „Beenden“

Ruft Sicherheitsabfrage auf und beendet anschließend das Programm. Wenn aktuell Sequenzen durchgeführt werden, wird die Fkt Abbruch ebenfalls aufgerufen

abbruch()

Ruft eine Sicherheitsabfrage auf und stoppt anschließend statusspezifisch alle laufenden Prozesse und gibt eine entsprechende Fehlermeldung aus. Zusätzlich wird für die Dauer des Abbruchprozesses die Statusanzeige angepasst.

Großteils werden bei Button-Click-Events Funktionen implementiert, auf welche bei den Click-Events verwiesen wird. Dies dient dazu, dass entsprechende Funktionen auch ohne Klick aufgerufen werden können. In diese Kategorie fallen folgende Funktionen.

messen\_fct(*boolean*)

Entsprechende Teile der GUI werden deaktiviert bzw. aktiviert und der Status wird angepasst. Die Klasse „messung“ (Kapitel 4.3.) wird instanziiert und die Funktion messen() (Seite 34) aufgerufen. Nach erfolgreicher oder abgebrochener Messung werden entsprechende Meldungen ausgegeben und die GUI angepasst.  
*Parameter:* WAHR für Aufruf bei automatischer Durchführung [AUTO]

analyse\_std\_fct()

kalibrieren\_fct()

ipf\_fct()

Entsprechende Teile der GUI werden deaktiviert bzw. aktiviert und der Status wird angepasst. Die Klasse „analyse“ (Kapitel 4.4.) wird instanziiert und inkludierte Funktionen - analyse\_std() (Seite 38), kalibrieren() (Seite 38) oder ipf() (Seite 44) - aufgerufen. Anschließend wird die GUI für weitere Anwendungen vorbereitet.

analyse\_spez\_fct(*boolean*)

Entsprechende Teile der GUI werden deaktiviert bzw. aktiviert und der Status wird angepasst. Die Klasse „analyse“ (Kapitel 4.4.) wird instanziiert und Funktion analyse\_spez() (Seite 43) aufgerufen. Nach erfolgreicher oder abgebrochener Analyse werden entsprechende Meldungen ausgegeben und die GUI angepasst.  
*Parameter:* WAHR für Aufruf bei automatischer Durchführung [AUTO]

auswertung\_fct(*boolean*)

Entsprechende Teile der GUI werden deaktiviert bzw. aktiviert und der Status wird angepasst. Die Klasse „auswertung“ wird instanziiert und Funktion auswerten() aufgerufen. Nach erfolgreicher oder abgebrochener Auswertung werden entsprechende Meldungen ausgegeben und die GUI angepasst.  
*Parameter:* WAHR für Aufruf bei automatischer Durchführung [AUTO]

auto\_fct()

Entsprechende Teile der GUI werden deaktiviert bzw. aktiviert und der Status sowie Endzeit der automatischen Durchführung angepasst. Die Form „auto\_auswahl“ (Kapitel 4.2.3.1) wird instanziiert und aufgerufen.

Eingaben, Speicherorte sowie benötigte Daten werden überprüft. Bei positivem Ergebnis werden, entsprechend der in „auto\_auswahl“ definierten Parameter, nacheinander die Funktionen - `messen_fct(true)`, `analyse_spez_fct(true)` für mittlere und lange Spektren sowie `auswertung_fct(true)` aufgerufen. Nach erfolgreicher oder abgebrochener Durchführung werden entsprechende Meldungen ausgegeben und die GUI angepasst.

Die automatische Durchführung ist nur sinnvoll, wenn die mittlere Messung der Proben bereits abgeschlossen ist und nun die lange Messung der Proben bevorsteht. Dies ermöglicht die anschließende automatische Analyse sowie Auswertung aller gemessenen Daten. Parameter der automatischen Durchführung sind mittels der Form `auto_auswahl` zu definieren und in Kapitel 6.1.8.1 näher beschrieben.

In Kapitel 6.1.8 sind umfangreiche Beschreibungen zur Verwendung der automatischen Durchführung angegeben, und der vollständige Programmcode ist zum besseren Verständnis im Anhang C angeführt. Inbegriffen sind in dieser Klasse auch alle Funktionen, die dem Benutzer dazu dienen Speicherorte, Analysedateien (.asf), Kalibrierdateien (.cal/.cnf), Peakdateien (.txt) und Spektren (.cnf) auszuwählen. Diese Routinen werden durch die Buttons „...“, neben den zugehörigen Textboxen, bzw. durch den Button „Spektren anzeigen“, ersichtlich in Abbildung 9, geöffnet.

#### 4.2.2.3. Fehlerbehandlung in der Stammklasse

Zur Fehlerbehandlung zählen in dieser Klasse alle Funktionen, welche Falscheingaben vom Anwender vermeiden.

Einerseits werden durch die Routinen

- `TBanzahl_KeyDown`
- `TBmessdauer_KeyDown`
- `TBenergie_KeyDown`
- `TBdateiname_KeyDown`

bei den ersten Dreien alle Tastaturtasten mit Ausnahme der Zahlen, Links, Rechts, Rückgängig und Entfernen, sowie die Modifiziertasten Shift, Strg, Alt und Alt Gr unterdrückt. Bei der vierten Routine werden nur Sonderzeichen mittels Modifiziertasten und Leerzeichen unterdrückt.

Andererseits werden bei der Auswahl von Daten Filter verwendet, welche die Auswahlmöglichkeiten des Benutzers auf benötigte Dateierendungen einschränkt. Dadurch werden dem Benutzer im Dateibrowser-Dialog auch nur passende Dateien angezeigt. Zu beachten ist, dass bei der Auswahl einer Kalibrierdatei, zwei mögliche Dateierendungen (.cal / .cnf) akzeptabel sind und der gewünschte Dateityp mittels Dropdown-Schaltfläche im Dateibrowser-Dialog gewählt werden kann.

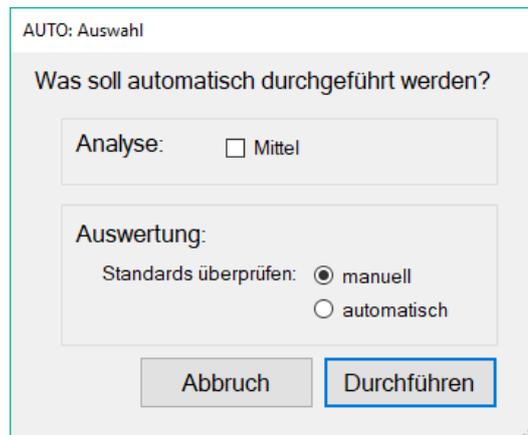
Bei der Auswahl der Abklingzeit-Vorwahl (mittel/lang) für Messungen, werden automatisch die Default Werte für Probenanzahl „50“ und Messdauern - lang: 10000s, mittel: 1800s - eingetragen, sowie Dateinamen mit den Attributen „lang“ bzw. „mittel“ spezifiziert.

## 4.2.3. Tochterklassen von DLFC\_index.vb

### 4.2.3.1. Klasse - auto\_auswahl.vb

Für die Auswahl der Parameter zur automatischen Durchführung wird eine neue Form „auto\_auswahl“, ersichtlich in Abbildung 10, erstellt.

Diese Form erscheint durch betätigen des Buttons [AUTO] nach erfolgreicher Überprüfung der Benutzereingaben. Die Spektren der mittleren Messung müssen im angegebenen Ordner verfügbar sein und mit der Datei probendaten.csv sowie mit den Benutzereingaben übereinstimmen. Genauere Informationen sind im Kapitel 6.1.8. zu finden.



**Abb.10:** GUI - Auswahlform für Parameter der automatischen Durchführung; Screenshot des Verfassers

Die Einstiegsfunktion der Klasse auto\_auswahl ist auto\_ausw\_fct(), welche ausgehend von DLFC\_index.auto\_fct() aufgerufen wird.

*auto\_ausw\_fct(boolean, boolean)*

Diese Funktion dient zur Definition der durzuführenden Aufgaben. Soll die Analyse der mittleren Messung durchgeführt werden oder sind die Reportdateien der mittleren Messung bereits vorhanden? Soll die Bearbeitung der Standards im Abschnitt Auswertung automatisiert oder manuell erfolgen?

*Parameter:* WAHR für Analyse der mittleren Spektren  
WAHR für autom. Bearb. der Standards

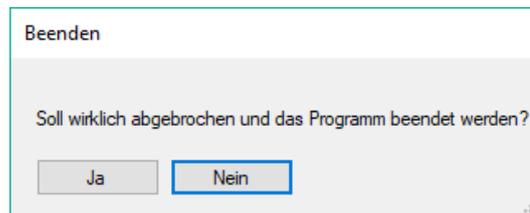
Die Funktionen Bgo\_Click und Bexit\_Click dienen bei Betätigung der Buttons für positive und negative Dialogantworten.

### 4.2.3.2. Klasse - control\_box.vb

Um Sicherheitsfragen beim Beenden des Programms und Abbrechen der Routinen zu implementieren, wird eine variable Benutzerabfrage - eine neue Form control\_box, dargestellt in Abbildung 11 - erstellt.

Diese dient als Sicherheitsstufe für unbeabsichtigtes Beenden oder Abbrechen der Routinen, wobei der Fokus der Form initial immer auf „nicht beenden“ gesetzt ist. In dieser Tochter-

klasse befinden sich nur zwei Funktionen - Byes\_Click() und Bno\_Click() - die bei Betätigung der zugehörigen Schaltflächen positive und negative Dialogantworten erzeugen.



**Abb.11:** GUI - Kontrollfrage für Beenden und Abbrechen der Routinen; Screenshot des Verfassers

## 4.3. KLASSE - MESSUNG

### 4.3.1. Problematik der Reproduzierbarkeit - Messklasse

Eine Messung ist dann als reproduzierbar zu bezeichnen, wenn zugehörige Parameter hinreichend gut dokumentiert sind. Diese Dokumentation beinhaltet einerseits fixe Parameter seitens des Detektors - Detektortyp, Probengeometrie und Korrektursystem und andererseits variable Parameter hinsichtlich der Proben - Kennzeichnung und Masseeinwaagen -, sowie der Messung - Messart, Messzeit und Messdauer.

Ein wichtiger Aspekt ist die Minimierung von Störeinflüssen und Manipulationen während der Messung. Die Folge ist die Automatisierung der Messung und des Probenwechsels, welche die Messung einer Probenreihe standardisiert. Durch die exakt gleiche Dauer jeder Messung und der gleichen Probengeometrie aufgrund des Probenwechslers sind Messdaten mit guter Qualität vergleichbar.

Dieses Programm ist an die Messsysteme PC1 und PC2 angepasst. Dementsprechend ändern sich Detektortyp, Korrekturmethode und Probengeometrie für Messungen mit Probenwechsler nicht und müssen für diese spezifische Anwendung nicht explizit für jede Messung angegeben sein.

Für die Verwendung der Messergebnisse in resultierenden Studien müssen die Spezifikationen des Systems ebenfalls bekannt sein und dokumentiert werden. Ebenso etwaige Veränderungen am Messsystem, wie Messungen ohne Probenwechsler mit veränderter Probengeometrie, müssen dokumentiert werden. Bei etwaigen Veränderungen der Messsysteme muss der Verlust der Vergleichbarkeit von Daten vor und nach der Veränderung in Betracht gezogen werden.

Die variablen Parameter der Proben und Messung ändern sich stets von Messung zu Messung. Diese müssen demnach für jede Probe explizit und am besten direkt im aufgenommenen Spektrum gespeichert oder im Dateinamen spezifiziert werden.

Im Zuge der Messung und Aufnahme des Spektrums werden etliche Parameter automatisch in der CAM-Datei (.cnf) gespeichert.

Einige davon lauten

- die Realtime - dies ist die tatsächlich gemessene Zeit, welche als Messzeit definiert wurde,
- die Livetime - dies ist die Zeit, in welcher effektiv Ereignisse detektiert wurden, also die resultierende Zeit bei Subtraktion der Totzeit des Detektors von der Realtime
- das Datum und die Uhrzeit der Messung.

Realtime und Livetime sind für korrigiertes Spektrum gleich, da bei der Loss-Free-Counting Korrekturmethode die Totzeit bereits berücksichtigt wird.

Probenreihennamen, Probennummer, Spektrentyp (LFC/UNC) und Messart (mittel/lang) werden automatisiert im Dateinamen zur eindeutigen Identifikation spezifiziert.

z.B.: 17G\_mittel\_0001\_LFC.CNF

Als zusätzliche Dokumentation werden die Parameter CAM\_T\_SDESC1 (536936537) und CAM\_T\_SDESC4(536936540) der CAM-Datei verwendet um die Arbeitsschritte für bessere Nachvollziehbarkeit zu referenzieren. (Kapitel 4.2.1.)

Für die Messung setzt sich die Referenz folgendermaßen zusammen.

aquisition - Datum [MMyy] - Spektrentyp [lfc/unc/dua]

z.B.: Q0318lfc/

### 4.3.2. Implementierung der Messklasse

Die Probenbehälter werden der Reihung der Probenliste angepasst im Probenwechsler angeordnet, wobei die erste zu messende Probe direkt mit dem Probenwechsler in den Detektor eingebracht wird. Dies dient dazu, den Detektor auf die erste Probe zu kalibrieren. Zur Kalibrierung des Detektors wird manuell Genie2000 geöffnet und die erste Probe ein paar Minuten gemessen. Anschließend wird mittels *Kalibrierung - Energie & Peakform* eine Kalibrierung durchgeführt (Kapitel 6.1.2.1.), welche direkt am Detektor gespeichert wird.

Für die Implementierung der Messung bedeutet dies, dass beim Start der Messung die Probe bereits im Detektor platziert ist und demnach der Probenwechsler vor der Messung der ersten Probe nicht aktiviert werden muss. Die Routine muss mit dem Start des Detektors eingeleitet werden.

Die Einstiegsfunktion der Klasse ist messen(), welche von DLFC\_index.messen\_fct() aufgerufen wird. Als Benutzereingaben werden Probenzahl, Messdauer, der gewünschte Speicherort sowie der Name der Probenreihe benötigt. Die Auswahlmöglichkeit für mittlere und lange Messungen bezieht sich auf die Abklingzeit zwischen Aktivierung der Proben und Zeitpunkt der Messung und ist nicht verpflichtend zu definieren. Wenn keine Auswahl getroffen wird, fehlt im Dateinamen die entsprechende Spezifikation und die Spektren werden nicht in Unterordner „mittel“ und „lang“ gegliedert.

Die Unterdrückung von Falscheingaben seitens des Anwenders ist teilweise in der Stammklasse DLFC\_index (Kapitel 4.2.2.3.) implementiert.

messen(*string, string*)

Nach Kontrolle der Benutzereingaben auf Vollständigkeit und Richtigkeit, werden die Subroutinen `create_directory_struct()` und `check_files()` aufgerufen. Die Gruppenbox „Aktuelle Messung“, ersichtlich in Abbildung 9, wird initialisiert und aktiviert. Angezeigt werden während der Messung immer aktuelle Probe, Status, Countdown der aktiven Messphase und Messende der gesamten Probenreihe. Anschließend wird eine Schleife gestartet, welche folgende Komponente durchläuft:

- Anpassen des Dateinamen der aktuellen Probe
- Starten der Messung mittels BATCH-Befehl „STARTMCA det:DLFC“ und Modifizierung „/REALPRESET“ [29]
- Aufrufen der Funktion `DLFC_index.sleep_fct`
- Dreimaliges Speichern des aufgenommenen Spektrums für *filename\_DUAL.CNF*, *filename\_LFC.CNF* und *filename\_UNC.CNF*
- Berechnung des korrigierten und unkorrigierten Integrals der gesamten Spektren sowie der korrigierten Livetime und der Incoming Count Rate (ICR)
- Anpassen bzw. eintragen der Parameter, Spektrengößen, Referenzen sowie Ergebnisse der durchgeführten Berechnungen
- Aufrufen der Subroutine `change_sample()` für Probenwechsel

Anschließend werden entsprechende Statusfelder zurückgesetzt und verborgen und die GUI angepasst.

*Parameter:*        Status „acq“ wird übergeben  
                      optionale Auswahl „mittel“ / „lang“ für Ordnerstruktur und spezifische Dateinamen

Die Änderungen der Parameter der CAM-Datei erfolgen durch eine Klasseninstanziierung der Bibliothek `CanberraDataAccessLib` und Aufruf der Funktionen `Open()`, `Close()`, `Save()`, `Spectrum()` und `Param()`. Die Syntax der Funktionen kann im Objektkatalog von Visual Studio eingesehen werden und ist in Anhang A angeführt.

Während des Ablaufes der Routine wird von jeder Probe Gesamtspektrum - DUAL - korrigiertes Spektrum - LFC - sowie unkorrigiertes Spektrum - UNC - gespeichert. DUAL wird in einen Unterordner „raw“ für Rohdaten gespeichert. An diesen Spektren wird während der Routine nichts verändert. Aus dem korrigierten Spektrum werden die korrigierten Peakflächen für die Auswertung extrahiert. Das unkorrigierte Spektrum ist für die Auswertung ebenfalls notwendig, da die Fehler der Integration der Peakflächen in diesem Spektrum nicht durch die LFC-Korrektur gestört werden. Daher werden die unkorrigierten Fehler verwendet und anschließend für die LFC-Korrektur korrigiert.

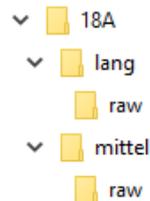
`change_sample()`

Diese Subroutine wird benötigt um zwischen den Messungen der einzelnen Proben, den Probenwechsler anzusteuern, damit die Probe im Detektor gewechselt wird. Dazu wird die externe Ressource an "C:\Program Files\AutoMeasure\_DLFC\DLFC\_ADV.exe" aufgerufen (Kapitel 3.2.2). Der Aufruf der Probenwechsler-Routine wird im Hintergrund ausgeführt. Der Status wird der aktuellen Messung angepasst.

`create_directory_struct(string, string, string)`

Für die geordnete Sicherung der Spektren wird eine für Probenreihenname und Speicherort spezifische Ordnerstruktur, ersichtlich in Abbildung 12, erstellt, insofern die betreffenden Ordner noch nicht vorhanden sind.

*Parameter:*      Pfad des gewünschten Speicherortes  
                    Dateiname der Probenreihe  
                    optionale Vorauswahl für mittlere und lange Messungen



**Abb.12:** Ordnerstruktur nach der Messung der Spektren „mittel“ und „lang“; Screenshot des Verfassers

Wird keine Vorgabe „mittel“ oder „lang“ ausgewählt, werden die Spektren sowie der Ordner „raw“ direkt in Stammordner - in Abb. 12: 18A - gespeichert. Die Unterordner „mittel“ und „lang“ werden in diesem Fall nicht erstellt.

#### 4.3.2.1. Fehlerbehandlung in der Messklasse

In dieser Klasse wird die Fehlervariable `error_num` der Stammklasse verwendet. Einerseits wird die Variable regelmäßig abgefragt für den Fall des manuellen Abbruchs der Messung durch den Anwender mittels des Abbruch-Buttons. Andererseits werden auch Rückgabewerte, beispielsweise zur Überprüfung der zu erzeugenden Spektren-Dateien - `check_files()`, mit der Fehlervariable gekoppelt. Dies dient dazu, mögliche Fehlerquellen vor dem Start der Messung zu beseitigen, damit die Messung reibungslos ausgeführt werden kann.

`check_files(integer, string, string)`

Mit dieser Routine werden alle Dateinamen auf bereits existierende und zu erstellende Dateien in den entsprechenden Ordnern überprüft. Dazu werden Dateinamen der gesamten Probenreihe mit den Spezifikationen „DUAL“, „LFC“ und „UNC“ erstellt und getestet. Wenn eine zu erstellende Datei bereits vorhanden ist, wird eine Fehlermeldung ausgegeben und anhand eines rückgegebenen Fehlerwertes die Messung nicht gestartet.

*Parameter:*      Probenanzahl  
                    Pfad des Speicherortes  
                    Dateiname der Probenreihe

Weiters werden Benutzereingaben auf Verwendbarkeit überprüft und zur Vermeidung von parallelem Starten von Subroutinen, werden die restlichen Teile der GUI für den Benutzer deaktiviert.

## 4.4. KLASSE - ANALYSE.VB

### 4.4.1. Problematik der Reproduzierbarkeit - Analyseklasse

Standardmäßig werden aktuell gemessenen Spektren mit der Analysesequenz „5 SIG-MA\_OR.asf“, abgelegt im Ordner C:\GENIE2K\CTFILES, analysiert. Mit der ebenfalls im gleichen Ordner abgelegten Reportvorlage „a\_heinz\_realtime.tpl“ wird eine Reportdatei mit allen gefundenen Peaks, deren Energien, Peakflächen, Fehler und einige weiteren Parametern, erzeugt. In der Reportdatei sind ebenfalls Datum und Zeit der Messung sowie der Analyse, Messdauer und der Pfad der Spektrendatei angegeben.

Die Analysesequenz „5 SIGMA\_OR“ beinhaltet die Schritte:

- Peaksuche - 2te Ableitung
- Peak-Fläche - Summe/Nicht-Lin. LSQ Fit
- Bericht

Die in Genie inkludierten Funktionen für Peaksuche und Peakfläche beinhalten die Möglichkeit der Veränderung einiger Parameter. Da aktuell die Analysesequenz auf das gesamte Spektrum mit denselben Parametern durchgeführt wird, werden einige Peaks, vor allem Peaks welche Teile von Multipletts sind, nicht gefunden und etliche Peaks schlecht gefittet. Dies führt zu falsch integrierten Peakflächen. Somit weichen einige von dieser Analyse erhaltenen Werte stark von den tatsächlichen Peakflächen ab. Im Zuge der Auswertung mussten die Peakflächen überprüft werden und etliche Peaks manuell gefittet werden, um die Werte nachbessern zu können.

Erstens führt die manuelle Nachbesserung mancher Werte von Peakflächen in Bezug auf die Reproduzierbarkeit auf eine nicht annehmbare, unmögliche Nachvollziehbarkeit weil Informationen fehlen welche Werte nachgebessert wurden, warum nachgebessert wurde und wie die korrekten Werte eruiert wurden. Zweitens existiert keine Form der Dokumentation, welche Analysesequenz verwendet und welche Parameter dafür definiert wurden.

Die Konsequenz daraus ist eine verbesserte Analyseverfahren und die Minimierung manueller Manipulation sowie eine nötige umfassende Dokumentation. Angestrebt wird eine jedem Nuklid beliebig genau angepasste Analyse und die Nachvollziehbarkeit der verwendeten Analyseparameter. Die Anpassung der Parameter erfolgt manuell und wird von Messung zu Messung sukzessive spezifischer und präziser.

Zusätzlich muss die Möglichkeit in Betracht gezogen werden, dass manche Nuklide schwer zu messen sind, und vereinzelt Proben auftreten können, welche nur manuell korrekt gefittet werden können. In diesen Fällen gewährleistet die inkludierte Dokumentation bei einem Interactive-Peak-Fit(IPF) die Nachvollziehbarkeit. Alle durchgeführten Maßnahmen zur Analyse werden zusätzlich mit Referenzen ausgestattet, um die Dokumentation zu vervollständigen.

## 4.4.2. Implementierung der Analyseklasse

Die Klasse Analyse umfasst alle Methoden, welche mit Spektren im Funktionsbereich von Genie durchgeführt werden können, bis zur Ausgabe von Reportdateien.

Inkludiert sind also die Gruppenboxen Kalibrierung und Analyse, welche in Abbildung 9 dargestellt sind. Diese umfassen die Prozeduren kalibrieren(), ipf(), analyse\_std(), analyse\_spez(), referenz(), report(), get\_peak\_table() und check\_files\_dir().

Für die Methoden werden einige Benutzereingaben benötigt, so müssen für jeden Arbeitsschritt die zu bearbeitenden Spektren ausgewählt werden. Dies erfolgt durch den Button „Spektren auswählen“ und ist in der Klasse DLFC\_index implementiert.

Zur Verfügung stehen die Methoden

- Kalibrierung
- Standardanalyse
- Spezialanalyse
- Interactive-Peak-Fit

### 4.4.2.1. Kalibrierung

Eine CAM-Datei(.CNF) beinhaltet etliche Parameter, welche der vorhandenen Kalibrierung zugeordnet sind. Diese sind aufgeteilt in die Blöcke Acquisition Parameters, Geometry Parameters, Energy/FWHM Calibration Results Parameters und Shape Calibration Results Parameters [28] und beinhalten enorm viele Records und Parameter.

Nach der Messung eines Spektrums, ist in der CAM-Datei die Kalibrierung, welche am Detektor gesichert wurde, hinterlegt. Optimalerweise sollte die erste Probe der Probenreihe vor dem Start der Messung manuell ansatzweise gemessen und kalibriert werden, damit die Kalibrierung am Detektor gespeichert werden kann. Die Kalibrierung setzt sich aus den Bereichen Energie- & Peakform- sowie Effizienzkalibrierung zusammen. Wichtig für diese Anwendung ist die Kalibrierung von Energie & Peakform, welche eine Beziehung zwischen den Kanälen des Spektrums und den Energieniveaus herstellt. Wird die Kalibrierung vor dem Messstart nicht durchgeführt, muss eine Möglichkeit bestehen, nach vollendeter Messung, mehrere Spektren automatisiert mit einer gewünschten Kalibrierung versehen zu können.

Dazu müssen die zu kalibrierenden Spektren, eine Kalibrierdatei und die zu übertragenden Teile der Kalibrierung ausgewählt werden(Kapitel 6.1.3.)

kalibrieren(*string()*, *string*)

Es wird überprüft, ob die ausgewählte Kalibrierdatei existiert. Die ausgewählten Blöcke der Kalibrierung - Energy&Peakform, Effizienz - werden von der Kalibrierdatei(.CAL oder .CNF) auf alle ausgewählten Spektren, mittels des BATCH-Befehls „MOVEDATA ‚Quelle‘ ‚Ziel‘“sowie den Modifizierungen /ECAL, /EFFCAL und /OVERWRITE übertragen [29].

Die betreffenden Statusfelder und die Aktivität von Teilen der GUI werden angepasst. Die Subroutine referenz() wird aufgerufen und abschließend eine entsprechende Meldung ausgegeben.

*Parameter:*        Array der Pfade aller ausgewählten Spektren  
                          Status der Durchführung

Die Referenzierung ist bei der Kalibrierung von folgender Form:

**Calibrate**        - ausgewählte Kalibrierdatei (.cal / .cnf)  
z.B.:    CGc50/

#### 4.4.2.2. Standardanalyse

Zur Durchführung einer Standardanalyse ist die Auswahl einer Analysesequenz notwendig. Vorgefertigte Sequenzen befinden sich im Ordner C:\GENIE2K\CTFILES. Die Erstellung individueller Analysesequenzen(ASE) wird in Kapitel (Kapitel 6.1.4.2.) näher erläutert.

Prinzipiell kann eine Analysesequenz beliebige Methoden enthalten, welche auch in *Genie2000 - Analyse* zu finden sind. Bemerkenswert sind die Möglichkeiten der Änderung der Vorlage(.tpl) für die Erstellung der Reportdateien sowie der Anpassung von Parametern zur spezifischeren Peaksuche oder Peakflächenintegration. Die Erstellung individueller ASE's eröffnet eine zweite Option, neben der Spezialanalyse, genauere Ergebnisse für die Integration der Peakflächen zu erhalten, als mit der Standardanalyse.

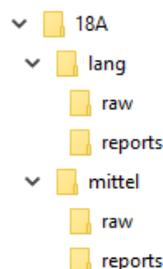
`analyse_std(string(), string)`

Statusfelder und GUI werden angepasst. Die Korrektheit der Analysesequenzbezeichnung wird überprüft und die Ordnerstruktur von Abbildung 12, wenn noch nicht vorhanden, erweitert für die Ausgabe der Reportdateien, ersichtlich in Abbildung 13. Darauf folgend wird die Existenz von Reportdateien mit gleichem Dateinamen kontrolliert und wenn nötig die Analyse abgebrochen, damit keine bestehenden Daten verloren gehen bzw. überschrieben werden. Innerhalb einer Schleife wird für jedes Spektrum

- die Funktion `referenz()` aufgerufen,
- die zugehörige CAM-Datei geöffnet,
- die Analyse durch Ausführen der Analysesequenz durchgeführt,
- die betreffende CAM-Datei geschlossen.

Die Statusfelder werden reinitialisiert und entsprechende Meldungen zur durchgeführten Analyse ausgegeben.

*Parameter:*        Array der Pfade aller ausgewählten Spektren  
                          Status der Durchführung



**Abb.13:** Ordnerstruktur nach der Analyse der Spektren „mittel“ und „lang“ ; Screenshot des Verfassers

Die CAM Dateien werden durch Instanziierung der Klasse CanberraDataAccessLib und Aufruf der Funktionen Open() und Close() geöffnet und gespeichert. Die Analysesequenz wird durch Instanziierung der Klasse CanberraSequenceAnalyzerLib, beziehungsweise durch Aufruf der implementierten Funktion Analyze() durchgeführt. Die Syntax der Funktionen kann im Objektkatalog von Visual Studio eingesehen werden und ist ebenfalls in Anhang A angeführt.

Die Referenz der Standardanalyse setzt sich wie folgt zusammen:

**Analyze** - ausgew. Analysesequenz (.asf)  
z.B.: A5 SIGMA\_OR/

Für die Ausgabe der Reportdateien werden der bestehenden, hauptsächlich verwendeten Reportvorlage „a\_heinz\_realtime.tpl“ einige Zeilen hinzugefügt, damit die Referenz in der Reportdatei angezeigt wird. Diese Vorlage wird unter dem Namen „a\_heinz.tpl“ gespeichert.

#### 4.4.2.3. Spezialanalyse

Die Spezialanalyse ist im Prinzip eine angepasste Analysesequenz, welcher der Struktur der Sequenz „5 SIGMA\_OR“ nachempfunden ist.

Analysiert werden pro Messung „mittel“ und „lang“ einige ausgewählte Nuklide, angeführt in den Tabelle 1 und 2 im Kapitel 2.1.2.4.

Problematisch sind die Nuklide  $^{153}\text{Samarium}$ ,  $^{177}\text{Lutetium}$ ,  $^{239}\text{Neptunium}$  und  $^{187}\text{Wolfram}$  bei der mittleren Messung sowie die Nuklide  $^{147}\text{Neodym}$ ,  $^{95}\text{Zirkonium}$ ,  $^{85}\text{Strontium}$  und  $^{58}\text{Kobalt}$  bei der langen Messung, wobei das Problem bei  $^{153}\text{Sm}$ ,  $^{177}\text{Lu}$ ,  $^{238}\text{Np}$ ,  $^{85}\text{Sr}$  von der Art ist, dass ein oder mehrere Peaks des Multipletts nicht signifikant genug sind, um vom Analysealgorithmus bei verwendeten Parametern gefunden zu werden. Dadurch wird das Multiplett falsch gefittet, und die Peakfläche ungenau integriert. Bei  $^{187}\text{W}$ ,  $^{147}\text{Nd}$ ,  $^{95}\text{Zr}$  und  $^{58}\text{Co}$  liegt das Problem im oft schwer aufzufindenden Peak, aufgrund eines schlechten Verhältnisses zwischen Peak-Intensität und Hintergrund.

Bei der Analysesequenz „5 SIGMA\_OR“ werden drei Analyseschritte, ersichtlich in Abbildung 14, nacheinander ausgeführt.

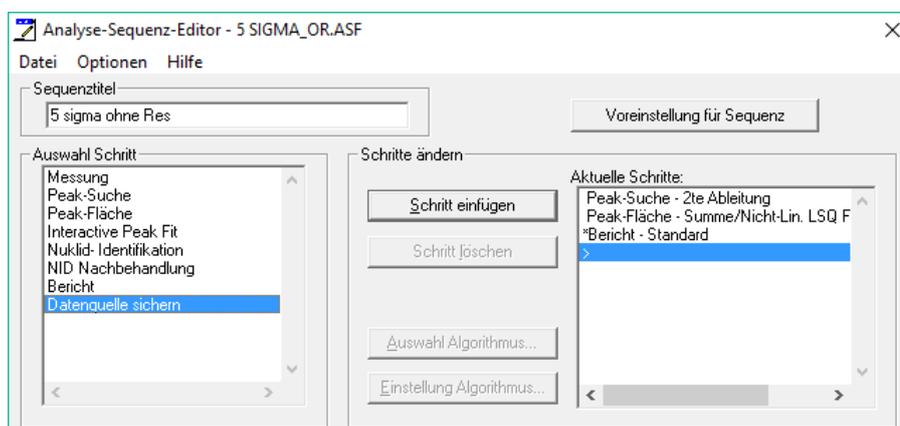
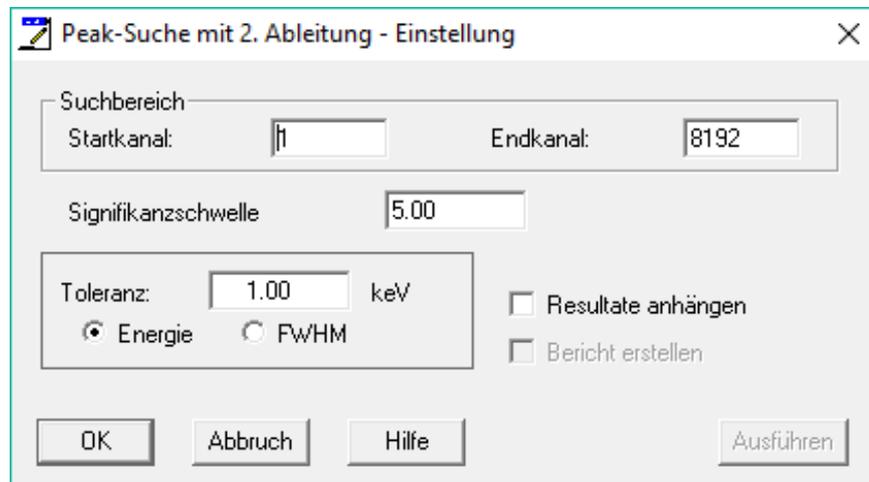


Abb.14: Analysesequenz „5 SIGMA\_OR“; Screenshot des Verfassers

Die Peaksuche „Peak-Suche mit 2. Ableitung“ wird für alle Peaks mit den gleichen Parametern, dargestellt in Abbildung 15, durchgeführt. Zu beachten ist der einstellbare Suchbereich mittels Startkanal und Endkanal, sowie der Parameter Signifikanzschwelle, welcher der entscheidenden Parameter für die Suche nach wenig signifikanten Peaks ist.



**Abb.15:** Parameter für Peak-Suche mit 2.Ableitung; Screenshot des Verfassers

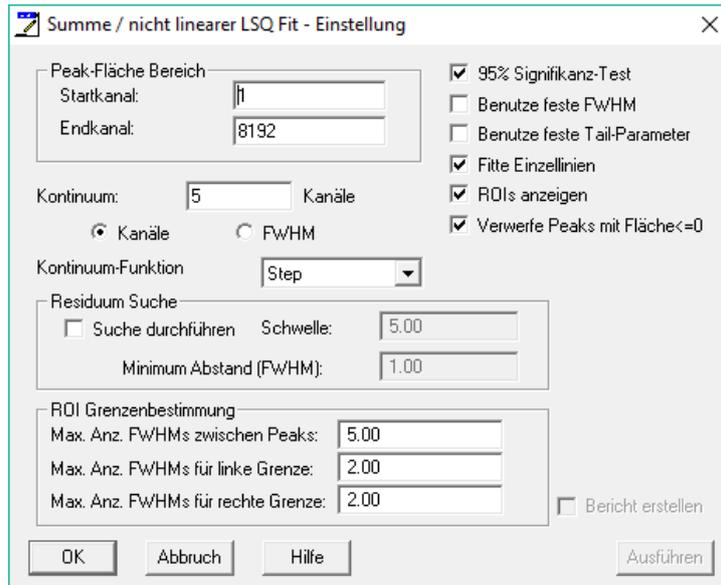
In Abbildung 16 werden die Parameter für den Analyseschritt der Peakflächenintegration „Summe/nicht linearer LSQ Fit“ angeführt. Auch in diesen Einstellungen besteht wieder die Möglichkeit zur Einschränkung auf Teile des Spektrums mittels Startkanal und Endkanal.

Weiters sind die Parametergruppe „Residuum Suche“ und die auswählbare Option „Fitte Einzellinien“ Parameter, welche für eine präzise Bestimmung der Peakflächen herangezogen werden. Die Residuum Suche ist vor allem zur Bestimmung der Anzahl der Peaks in Multipletts und die damit verbundene Genauigkeit der Peakflächen-Integration des zu analysierenden Peaks wesentlich. Der Parameter „Fitte Einzellinien“ bedeutet, dass Einzelpeaks gefittet werden. Ist diese Checkbox nicht gesetzt, werden alle Zählraten der dem Peak betreffenden Kanäle summiert.

Die Vorteile der einfachen Summation bei Einzellinien sind:

- Der Fehler der Linienflächenbestimmung rührt einzig vom Fehler der Zählrate, welcher nicht wie bei der Fit-Methode durch den Fehler der Anpassung vergrößert wird.
- Die Rechenzeit erhöht sich durch die Anpassung um einen Faktor > 10 [15]. Dementsprechend ist die einfache Summation effizienter (da keine Anpassung durchgeführt wird).

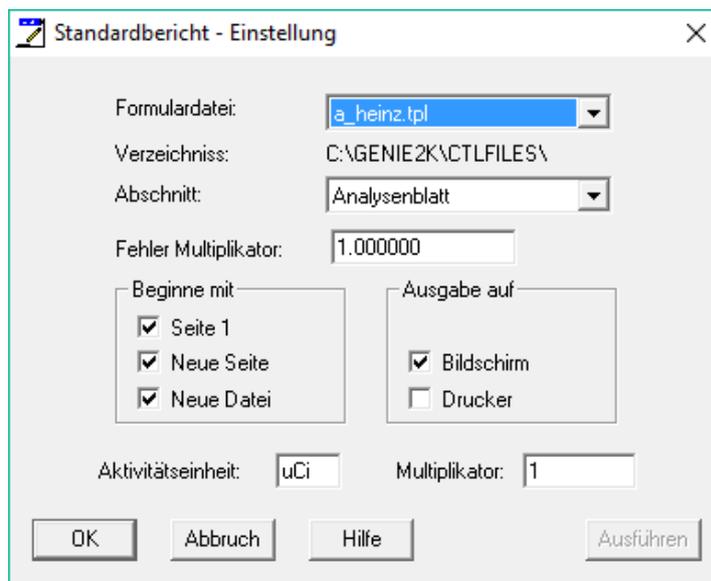
Dementsprechend sollte hauptsächlich die Methode der Summation benutzt, und nur bei Energielinien, deren Grenzen mit benachbarten Linien überlappen oder bei nicht gleichmäßigem Hintergrund, auf die Fitmethode zurückgegriffen werden [15]. Wird die Summationsmethode benutzt, fließt die „Form“-Kalibrierung nicht ins Ergebnis ein, da für die Berechnung der Peakfläche kein Fit verwendet wird.



**Abb.16:** Parameter für Peak-Flächen-Integration - „Summe /nicht lin. LSQ FIT“ ; Screenshot des Verfassers

Werden für Anfangs- und Endkanal der Peaksuche oder Peakflächenintegration zu kleine Bereiche gewählt, wird der Hintergrund falsch angepasst und der Peak falsch gefittet. Werden die Bereiche zu groß gewählt, werden mehrere Peaks als Multiplett mit den gleichen Analyseparametern analysiert. Dies führt wiederum dazu, dass nur teilweise an Energielinien angepasste Parameter verwendet werden. Für zu hoch angesetzte Signifikanzschwelle, werden Peaks mit darunterliegender Signifikanz nicht gefunden. Für zu hoch angesetzte Schwelle der Residuum-Suche werden Peaks eines Multipletts, welche geringere Residuen zur Folge haben, nicht als eigenständiger Peak erkannt.

Um die analysierten Daten in Reportdateien auszugeben, wird die Methode „Bericht - Standard“ verwendet, mit in Abbildung 17 angeführten Variationsmöglichkeiten.



**Abb.17:** Standardbericht - Einstellungsmöglichkeiten; Screenshot des Verfassers

Intensitäten von Peaks bzw. Strukturen von Multipletts sind für den Großteil der Proben für gleiche Nuklide sehr ähnlich. Daher liegt die Vorgehensweise nahe, an die Nuklide angepasste Analysen zu definieren. Definiert werden dabei die erwähnten Parameter der Peak-suche und Integration der Peakfläche in einer Textdatei, gespeichert im Standard-Stammverzeichnis C:\GENIE2\CTFILES\peaks.txt, welches durch den Benutzer geändert werden kann. Die Parametertabelle ist folgend angeführt:

energie	kanal_anf	kanal_end	schwelle	signifikanz(2.abl)	FIT
<b>mittel:</b>					
103,2	241	280	5	1	0
208,4	532	551	2	1	0
277,4	713	747	2	1	0
559,1	1427	1466	1,5	2	0
684,7	1773	1794	2	2	1
1369	3536	3561	5	5	1
1525	3941	3966	5	5	1
1596	4126	4152	5	5	0
2754	7129	7159	5	5	1
<b>lang:</b>					
145,4	309	384	5	5	0
177,2	453	467	5	5	1
208,4	533	547	3	2	1
312,2	793	817	5	5	1
320,1	822	838	3	3	1
482,2	1227	1257	5	5	1
496,3	1280	1296	3	2	1
514	1315	1342	3	1	0
531	1371	1383	2	1	1
756,7	1956	1991	3	1	0
795,9	2050	2113	2	1	0
879,4	2274	2288	5	5	1
1077	2781	2822	3	3	0
1099,2	2839	2861	5	5	1
1115,5	2875	2916	5	5	0
1173,2	3021	3066	5	5	0
1189	3072	3094	3	2	1
1221,3	3139	3203	5	5	0
1408	3620	3663	5	5	1
1691	4380	4392	2	2	0

**Tab.4:** Parametertabelle - peaks.txt - für Spezialanalyse

Die Auswahl der Parameter erfolgte hauptsächlich durch Untersuchung der einzelnen Peaks, deren Signifikanz und Form, sowie der Untersuchung von Multipletts. Die günstigeren Parameter wurden dann durch Vergleich der Analyse- und Auswertungsergebnisse (Kapitel 5) nach der Methode Trial&Error eingeschränkt. Die Spezifikation, ob Einzellinien gefittet werden sollen, wird ebenfalls danach entschieden, mit welchen Einstellungen die Abweichungen von alter zu neuer Auswertung minimiert werden.

Diese Parameter werden standardmäßig für mittlere und lange Spezialanalysen verwendet, können aber für außergewöhnliche Probenreihen verändert werden. Nach erfolgter Spezialanalyse wird die verwendete Parametertabelle aus dem Stammverzeichnis zu den erzeugten Reportdateien kopiert.

`analyse_spez(string(), string, boolean)`

Benötigte Benutzereingaben sind die Parametertabelle(.txt) und Wahl der Spektrenart - „mittel“ oder „lang“. Default-Werte sind „C:\GENIE2K\CTLFILLES\peaks.txt“ und „mittel“. Nach Überprüfung der Parametertabelle, Anpassung der Statusfelder und Aufruf der Funktion `check_files_dir()` wird die Parametertabelle durch Aufruf von `get_peak_table()` eingelesen.

Der weitere Programmablauf ist getrennt durch die Vorwahl „mittel“ und „lang“, ist aber weitgehend, abgesehen von Variablenbezeichnungen und Variableninhalt, strukturell ident. Zuerst wird die Statusanzeige des Automatik-Modus bei deaktiviertem „AUTO“ als Analysenendanzeige verwendet und die Spalten der eingelesene Parametertabelle in simple, nutzbare Arrays - peaks, KA, KE, schwelle, signifikanz und fit - geteilt.

Danach wird eine Analyseschleife für jedes Spektrum und jede einzelne Energie in der Parametertabelle mit folgenden Schritten gestartet:

- Peaksuche mit BATCH-Befehl „PEAK\_DIF ‚Quellspektrum‘“ mit den Modifizierungen /CHANNELS und /SIGNIF [29]
- Integration der Peakfläche mit BATCH-Befehl „AREA\_NL1 ‚rum‘“ sowie den Modifizierungen /RES\_SEARCH, /RES\_THRESHOLD, /RES\_SEPARATION und /FIT [29]
- Aufruf der Subroutine `referenz()` (Seite 45),
- Aufruf der Subroutine `report()` (Seite 45),
- Kopie der Reportdateien von Standardordner C:\GENIE2K\REPFILLES in den Zielordner „reports“

Anschließend wird die verwendete Parametertabelle ebenfalls in den Zielordner kopiert, und Statusanzeigen sowie GUI angepasst.

*Parameter:*       Array der Pfade aller ausgewählten Spektren  
                  Status der Durchführung  
                  Status des Automatik-Modus - ON/OFF

Die Referenz der Spezialanalyse hat folgende Form:

**Special Analyze**       -   ausgew. Peaktabelle (.txt)       -   Typ [M/L]  
z.B.:   SApeaksM/

#### 4.4.2.4. Interactive-Peak-Fit (IPF)

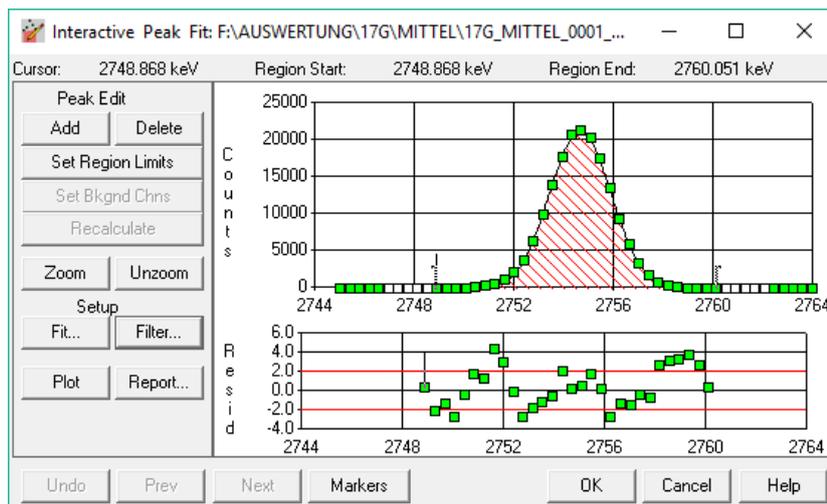
Da oft Proben natürlichen Ursprungs gemessen werden, sind Ungenauigkeiten bei der Analyse nicht auszuschließen. Bei derartigen Ausnahmen kann mittels IPF ein falsch analysierter Peak nachträglich manuell gefittet werden und die Werte der Peakfläche und des Fehlers in der danach erzeugten Reportdatei ausgetauscht werden. Dies soll ebenfalls im Sinne der Reproduzierbarkeit dokumentiert werden. Hierfür besteht die Möglichkeit nach dem IPF eine individuelle Referenz durch eine Benutzereingabe hinzuzufügen.

Für die Durchführung des IPF müssen das betreffende Spektrum bzw. die betreffenden Spektren ausgewählt werden. Optional kann eine Energie als Filter spezifiziert werden, an welcher das Spektrum im IPF-Fenster angezeigt werden soll. Dies dient zur effizienteren Bearbeitung, wenn ähnliche Probleme bei gleichen Energien und mehreren Spektren auftreten. Der Aufruf der Funktion `IPF()` führt beispielhaft zur Abbildung 18 und anschließend zur Eingabemöglichkeit einer individuellen Referenz (Kapitel 4.4.3.).

ipf(string())

Die Klasse input\_box wird instanziiert. Innerhalb einer Schleife wird für jedes ausgewählte Spektrum der Interactive-Peak-Fit mittels des BATCH-Befehls „IP-FIT ‚Quellspektrum‘“ und Modifizierung /ENERGY [29] - wenn eingegeben - aufgerufen. Anschließend wird die Funktion eingabe() der Klasse input\_box (Kapitel 4.4.3.) aufgerufen und die resultierende Eingabe an die Funktion referenz() übergeben.

*Parameter:*        Array der Pfade aller ausgewählten Spektren



**Abb.18:** Beispiel - Interactive-Peak-Fit Aufruf; Screenshot des Verfassers

Die Referenz des IPF setzt sich mit inkludierter, individueller Benutzereingabe wie folgt zusammen:

**Interactive Peak Fit**    -    Benutzereingabe nach IPF  
z.B.: ipf208,559/

#### 4.4.2.5. Allgemeine Funktionen

referenz(string, string)

Das betreffende Spektrum wird geöffnet, und die bereits in den Parametern CAM\_T\_SDESC1 und CAM\_T\_SDESC4 gespeicherte Referenz ausgelesen. Die neue Referenz wird hinzugefügt und überprüft ob die Referenz mehr als 64 Zeichen aufweist. In diesem Fall wird der Parameter CAM\_T\_SDESC4 ebenfalls zur Speicherung verwendet.

*Parameter:*        Pfad des aktuellen Spektrums  
                         Anzufügende Referenz

Diese Funktion wird aus den Routinen Kalibrieren() (Seite 38), Analyse\_std() (Seite 38), Analyse\_spez() (Seite 43) und IPF() (Seite 44) aufgerufen. Die Referenzen sind sehr kurz gehalten, da ein Parameter nur 64 Zeichen beinhalten kann. Mit einer wahrscheinlichen Abfolge - Messung(9) - Spezialanalyse(11) - Kalibrierung(~10) - Spezialanalyse(11) - werden bereits 41 von verfügbaren 64 verbraucht. Da IPF-Referenzen individuell erstellt werden können, wird bei einer Überschreitung von 64 Zeichen der Parameter CAM\_T\_SDESC4 ebenfalls verwendet. Dadurch stehen für die Referenzierung 128 Zeichen zur Verfügung. Aus diesem Grund sind individuelle Referenzen im Zuge des IPF kurz zu halten.

report(*string, integer, integer*)

Die Sektionen der Reportvorlage „Header“, „Tabelle“ und „Fehler“ werden definiert. Die Sektion „Header“ wird nur bei der Analyse der ersten Energie der Parametertabelle ausgegeben. Die Sektion „Tabelle“ wird pro Analyse eines Peaks der Reportdatei angehängt. Daraus resultiert die Tabelle, deren Daten zur Auswertung herangezogen werden. Die Sektion „Fehler“ wird nur bei der Analyse des letzten Eintrags der Parametertabelle angehängt und dient als Terminator.

*Parameter:*           Pfad des aktuellen Spektrums  
                          Index der Zeile der Parametertabelle  
                          Index der Schlusszeile der Parametertabelle

Für die Spezialanalyse wird eine neue Vorlage „sanal.tpl“ erstellt. Dazu wird die Vorlage „a\_heinz.tpl“ als Vorlagenvorlage benutzt und die benötigten Sektionen „Header“, „Tabelle“ und „Fehler“ eingefügt, damit aus diesen Sektionen verwendbare Reportdateien kreiert werden können.

Die Reportdateien werden durch Instanziierung der Klasse CanberraReporterLib und Aufruf der Funktion ExecuteReport() erstellt und erweitert. Die Syntax kann im Objektkatalog von Visual Studio eingesehen werden und ist ebenfalls in Anhang A angeführt.

get\_peak\_table()

Diese Funktion dient nur zum Einlesen der Parametertabelle - peaks.txt - nicht zur Weiterbearbeitung des Inhalts bzw. zur Übertragung in Variablenarrays. Dazu wird eine Instanz der in VB inkludierten Streamreader-Klasse erstellt und als Objekt definiert. Dieses Objekt wird geöffnet, alle nichtleeren Zeilen der Datei ausgelesen und in eine Array Liste, zur Rückgabe an die Funktion Analyse\_spez(), gespeichert.

#### **4.4.2.6. Fehlerbehandlung in der Analyseklasse**

Die Fehlervariable error\_num wird durch Rückgabewerte von der Funktion check\_files\_dir() gesetzt und anschließend überprüft. Dadurch wird die Analyse vor Beginn der Analyseschleife bei Auftreten eines Fehlers beendet. So wird vermieden, dass Reportdateien bereits ausgewerteter Spektren überschrieben werden.

check\_files\_dir(*string()*)

Es wird das Zielverzeichnis der Ausgabedateien - \reports - überprüft und bei Nichtexistenz neu erstellt. Die Dateinamen aller zu erstellenden Ausgabedateien, Reportdateien für jedes analysierte Spektrum, werden auf Existenz überprüft. Wenn nötig wird die Analyse abgebrochen, um Datenverlust durch Ersetzen von Dateien zu vermeiden.

*Parameter:*           Array der Pfade aller ausgewählten Spektren

Mehrere Sicherheitsabfragen werden zum Beispiel für Formatierungsfehler der Parametertabelle implementiert. Diese sind mit entsprechenden Fehlermeldungen verknüpft.

### 4.4.3. Tochterklasse von Analyse.vb

Aufgerufen wird die Tochterklasse `input_box.vb` mittels der Einstiegsfunktion `eingabe()` von der Funktion `ipf()` (Seite 44) und dient dazu, nach dem Interactive-Peak-Fit eine durch den Benutzer individuell angepasste Referenz hinzuzufügen. Die GUI der Eingabebox ist in Abbildung 19 dargestellt, und beinhaltet die Information welches Spektrum bearbeitet wird, sowie eine beispielgebende Anleitung.

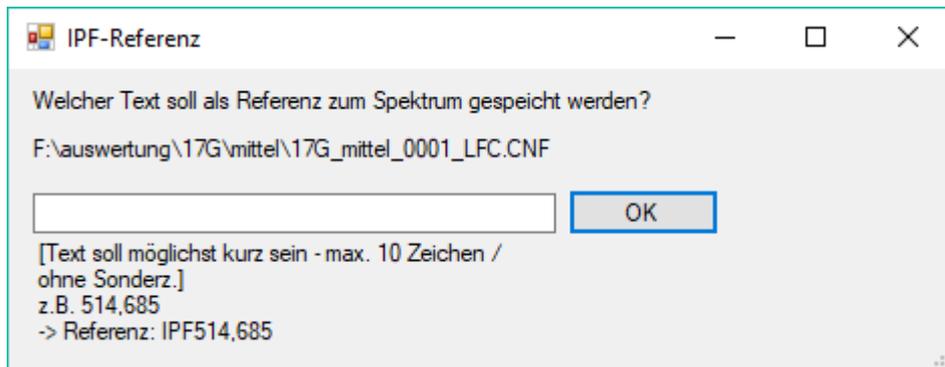


Abb.19: Referenz-Eingabemöglichkeit nach IPF; Screenshot des Verfassers

`eingabe(string)`

Die Funktion reinitialisiert Referenzvariablen und passt die Benutzeroberfläche dem aktuellen Spektrum an. Die Form `input_box` wird aufgerufen und nach einer erfolgreichen Benutzereingabe wird die anzuhängende Referenz zurückgegeben.

*Parameter:* Pfad des aktuellen Spektrums

Die Funktion `Bok_Click` dient zur Erstellung einer positiven Dialogantwort beim Aufruf der Form `input_box`.

## 4.5. KLASSE - AUSWERTUNG.VB

### 4.5.1. Problematik der Reproduzierbarkeit - Auswertungsklasse

Die Daten der Reportdateien der LFC-Spektren werden aktuell mittels einer Visual Basic Routine in eine Excel Datei „NAA\_Auswertung.xls“ eingespeist. Diese Excel Datei beinhaltet alle benötigten Daten und Berechnungen für die folgende Auswertung. Wie bereits in Kapitel 4.4.1 erwähnt, führen fehlerhaft analysierte Peaks zu ungenauen und teilweise fehlenden Peakflächen und Fehlern bei der Auswertung. In besagter Excel-Datei werden fehlerhafte Werte von Peakflächen und Fehler durch manuellen Fit der entsprechenden Peaks nachgebessert. Die Änderung dieser Daten ist kaum reproduzierbar. Dieses Problem von nötigen, nachträglichen Änderungen von Werten wird durch die Anpassung der Parameter der Analysemethode an einzelne Energien von Nukliden, ersichtlich in Tabelle 4, weitestgehend minimiert. Notwendige Änderungen können durch die Funktion `IPF()` (Seite 44) durchgeführt und dokumentiert werden.

Ein weiteres Problem der Nachvollziehbarkeit tritt durch qualitative Anpassungen während der Auswertung auf. Im speziellen werden einige Standards, Proben mit bekannter Zusammensetzung, bei jeder Probenreihe mitgemessen. Diese Standards dienen bei der Berechnung der Nuklidkonzentration als Referenz, welche Konzentration eines Nuklides in einer Probe welche Zählrate zur Folge hat. Damit dieser Referenzwert eine hohe Qualität aufweist, müssen mehrere Standards mitgemessen werden und Mittelwerte sowie Standardabweichung für jede Energielinie berechnet werden. Durch unterschiedlich stark ausgeprägte Peaks von Energien im Spektrum, variiert die Qualität der Messung und damit der Fehler der Peakfläche. Dadurch entsteht eine Verteilung der Standard-Referenzwerte, welche durch die Standardabweichung ersichtlich wird. Anhand des Kriteriums, die Standardabweichung dürfe nicht größer 5 % sein, werden sukzessive Standard-Referenzwerte, welche vom Mittelwert stark abweichen manuell oder automatisch entfernt.

Die Entfernung von Standard-Referenzwerten muss ebenfalls dokumentiert werden, damit die Auswertung nachvollziehbar ist. Da während der Auswertung alle Reportdateien ausgelesen werden, sind auch das Auslesen und die Speicherung der ID-Referenzen als übersichtliche Logdatei in diesem Kapitel implementiert. Manuelle Manipulationen der Berechnungen werden durch die Automatisierung der Auswertung nahezu eliminiert. Unvermeidbare manuelle Eingriffe werden automatisch und vollständig dokumentiert.

#### 4.5.2. Implementierung der Auswertungsklasse

Für die Abwicklung der Auswertung sind einige Daten zur Verfügung zu stellen, die größtenteils mittels der Dateien probendaten.csv, nuklide\_mittel.csv und nuklide\_lang.csv bereitgestellt werden. Die restlichen benötigten Daten, wie Peakflächen, Fehler, Messzeit und Abklingzeit, werden aus den Reportdateien der analysierten Spektren ausgelesen.

**probendaten.csv** beinhaltet eine Liste aller Proben der Probenreihe. Ein Auszug ist in Tabelle 5 ersichtlich und enthält die Bezeichnungen der Proben, die Masse der Einwaagen in Milligramm sowie die Definition, ob eine Probe als Standard auszuwerten ist. Der Wert „0“ in Spalte „standard“ bedeutet, dass die Probe nicht als Standard ausgewertet wird, wobei bei allen Standards die korrekte Bezeichnung der Standards, wie in den folgenden Dateien angeführt, verwendet werden muss, damit die Konzentrationen der Standards korrekt verknüpft werden. Diese Datei muss für jede Probenreihe manuell erstellt werden und muss beim Starten der Auswertung im auszuwertenden Ordner - z.B. F:\auswertung\17G\auswertung - abgelegt sein.

1	name	masse[mg]	standard
2	CFA	118,09	CFA
3	LSS	150,05	LSS
4	GBW	125,57	GBW
5	SO-1	135,42	SO-1
6	IMS	104,78	IMS
7	L651-02	125,78	0
8	L651-03	119,02	0

Tab.5: Auszug aus Eingabedatei - probendaten.csv

**nuklide\_mittel.csv** und **nuklide\_lang.csv** sind ebenfalls in Tabellenform verfügbar und beinhalten alle Elemente bzw. deren ausgewählte Energielinien, welche ausgewertet werden sollen. In der ersten Spalte der Tabelle ist das Element ersichtlich, dessen Konzentration bestimmt werden soll. Die folgenden drei Spalten beinhalten Halbwertszeit, Energie und Energieintervall der Radionuklide, deren Peak zur Analyse und Auswertung des Elements herangezogen wird. Das Energieintervall bezeichnet in welchem Energiebereich die Reportdateien nach Peaks durchsucht werden sollen. Alle weiteren Spalten sind mit den Konzentrationen der Elemente in den verfügbaren Standards gefüllt und in der Einheit Mikrogramm pro Gramm Probe (parts-per-million - ppm) angegeben.

Sollen zusätzliche Peaks ausgewertet werden, müssen diese in dieser Datei angefügt werden. Günstig wäre ebenfalls eine Anfügung der entsprechenden Energielinien und Parameter in der Parametertabelle peaks.txt für eine spezifische Analyse der neu hinzukommenden Energielinien.

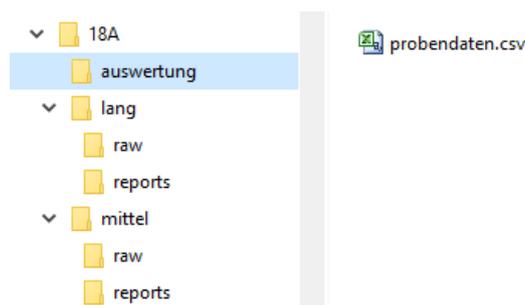
Beispielhaft ist die Datei nuklide\_mittel.csv in Tabelle 6 angeführt. Da im Allgemeinen dauerhaft die gleichen Nuklide bzw. Energielinien ausgewertet werden, sind diese Dateien in C:\GENIE2K\CTLFILLES\ gespeichert. Nach Ende der Auswertung werden diese Dateien in den Stammordner der Probenreihe zur Dokumentation kopiert.

Nuklid	HWZ (s)	Energie (keV)	E_interval	SO-1	CFA	LSS	GBW	SAT5	Ferrosilokon	IMS	Bonn
Sm	166572	103,2	1	7,9	20	6,45	11,7	5,946		10,8	6,13
Lu	581818	208	0,7	0,31	1,2	0,45	0,67	0,74			0,468
U	203602	277,4	1	1,71	8,79	3,62	4,83	5,664		10,4	5,3
As	93121,2	559,1	1	2	136,2	17,41	0,66	2,751		45,3	38,2
W	86040	685,8	1,1		5,6	1,2	1,1			6,2	5,08
Na2	53852	1369	1	20030	2010	7200	19065,9	36384	1700	6810	1960
K	44496	1525	1	26399	19500	20010	45077,55	25724		20540	13370
La	144986	1596	1	54	94	35,13	82,7	30,23		73,5	48,94
Na	53852	2754	1	20030	2010	7200	19065,9	36384	1700	6810	1960

**Tab.6:** Eingabedatei - nuklide\_mittel.csv

Der auszuwertende Ordner - in Abbildung 20 der Ordner „18A“ - muss vom Benutzer ausgewählt werden. Da die Datei probendaten.csv vor dem Start der Auswertung darin vorhanden sein muss, ist vom Anwender die Ordnerstruktur zu erweitern und die Datei korrekt abzulegen.

Zusätzlich muss vom Anwender durch Setzen der Checkbox „Standards bearbeiten: Automatisch“ definiert werden, ob die Standards automatisch oder vom Benutzer manuell bearbeitet werden sollen.



**Abb.20:** Ordnerstruktur vor Beginn der Auswertung; Screenshot des Verfassers

Alle Ausgabedateien, welche durch die Auswertung konstruiert werden, werden automatisch im Ordner „auswertung“ erstellt. Dies sind die Dateien *PRN\_logfile.csv*, *PRN\_messwerte.csv*, *PRN\_standards\_init.csv*, *PRN\_standards\_fin.csv* und *PRN\_auswertung.csv*. PRN steht für Probenreihenname. Nähere Beschreibungen zu den einzelnen Ausgabedateien werden in Kapitel 6.1.1. gegeben.

#### 4.5.2.1. Einstiegsfunktion - auswerten()

Die Einstiegsfunktion der Auswertung-Klasse ist `auswerten()` und wird von der Stammklasse aufgerufen.

`auswerten(string)`

Zu allererst wird der ausgewählte Ordner übergeben und durch Aufruf der Funktionen `check_dir()` (Seite 57) und `check_files()` (Seite 57) werden Ordner und Dateien überprüft. Anschließend wird die Datei *PRN\_logfile.csv* erstellt, wenn zugehörige Spektren in den entsprechenden Ordnern vorhanden sind. Die benötigten Dateien werden mittels der Funktionen `get_probendaten()` (Seite 54) und `get_nuklide_data()` (Seite 54) eingelesen. Danach wird die Anzahl der auszuwertenden Proben und die Anzahl der Energielinien für die Spektren der mittleren und langen Messung definiert, um entsprechende Ergebnistabellen zu konstruieren. Anschließend werden die Routinen `evaluation_mittel()` (Seite 50) sowie `evaluation_lang()` (Seite 50) aufgerufen. Retournieren diese Routinen positive Rückgabewerte, werden die Ausgabedateien *PRN\_messwerte.csv* sowie *PRN\_auswertung.csv* konstruiert und durch Aufruf der Subroutinen `write_line()` und `write_res()` gefüllt.

*Parameter:*                      Status der Methode

Die beiden Funktionen `evaluation_mittel()` und `evaluation_lang()` sind strukturell ident, sind aber in zwei Funktionen separiert, da sich die Quelldaten in unterschiedlichen Ordnern befinden und durch die schrittweise Ausführung eine komponentenbezogene Fehlerüberprüfung implementierbar wird.

`evaluation_mittel(string(,),string(,), string, string(,), string(,), string)`

`evaluation_lang(string(,),string(,), string, string(,), string(,), string)`

Variablen werden definiert. Aus *probendaten.csv* wird eine Liste der Positionen der Standards in der Probenreihe herausgefiltert. Reportdateien werden eingelesen und deren Anzahl kontrolliert. Ergebnistabellen werden dimensioniert, und die erste Auswertungsschleife wird mit folgenden Komponenten durchlaufen:

- Pfade der LFC-Reports und UNC-Reports der aktuellen Probe werden definiert und Daten werden durch Aufruf der Funktionen `get_rpt_table()` (Seite 55), `get_acqstarttime()` (Seite 56) und `get_acqduration()` (Seite 56) eingelesen.
- Daten der beiden Reportdateien werden aufbereitet, indem Daten von auszuwertenden Energielinien separat gespeichert werden - Peakflächen werden aus dem LFC-Bericht und Fehler aus dem UNC-Bericht extrahiert.
- Fehler werden mit (9) korrigiert.

- Nach Definition der Abklingzeit wird die erste Berechnung durch den Funktionsaufruf `calculate_1()` (Seite 52) durchgeführt und anschließend Messwertdaten und Ergebnisse der ersten Berechnung für alle Proben in einer Tabelle zusammengefasst.

Im zweiten Schritt werden mittels der Funktion `calculate_std()` (Seite 53) die R-Werte der Standards und zusätzlich der Mittelwert der R-Werte, sowie die Standardabweichung, der Fehler des Mittelwerts-Absolut, der Fehler des Mittelwerts-Relativ und das Fehlerquadrat des Mittelwerts-Relativ berechnet. Diese initialen „Standard-Werte“ werden ohne manuelle oder automatische Manipulation in `PRN_standards_init.csv` dokumentiert.

Anschließend werden die Standard-R-Werte, abhängig von der Benutzerauswahl, manuell oder automatisch überprüft. Ziel ist es, eine Standardabweichung des Mittelwerts der R-Werte zu erreichen, welche den Grenzwert von 5 % unterschreitet, um eine möglichst akkurate Berechnung der Nuklidkonzentrationen zu ermöglichen.

Bei manueller Bearbeitung der Standards wird eine Instanz der Klasse `standards.vb` (Kapitel 4.5.3) initialisiert. Für jedes Nuklid wird innerhalb einer Schleife

- die GUI der Form „standards“ mit entsprechenden Standardwerten angepasst und aufgerufen,
- von positiv bzw. negativ abhängigen Rückgabewerten der Form, die Standardwerte den Benutzereingaben der Standards-Form angepasst oder die Auswertung abgebrochen,
- die Entfernung von Werten durch den Benutzer in `PRN_standards_fin.csv` dokumentiert, damit diese nachvollziehbar sind.

Danach werden der Mittelwert und Fehler der Standards, mit `recalculate_std()` (Seite 53) und den modifizierten R-Werten, neu berechnet. Die finalen Werte der Standard-Berechnung werden in `PRN_standards_fin.csv`, durch `write_str_tbl()` (Seite 56) festgehalten.

Wird vom Benutzer die automatische Bearbeitung der Standards gewählt, werden Variablen definiert und initialisiert und eine Schleife eröffnet, die pro Nuklid

- die aktuelle Anzahl der R-Werte definiert und
- nach Überprüfung der Kriterien eine weitere Schleife initialisiert,
  - welche den am weitesten vom Mittelwert abweichenden Wert definiert und löscht,
  - die Standard-Berechnungen mittels `recalculate_std()` erneut durchführt,
  - die Änderungen an R-Werten ebenfalls zur erhöhten Reproduzierbarkeit in `PRN_standards_fin.csv` dokumentiert
  - und die neue Anzahl der R-Werte definiert.

Diese Schleife wird so lange ausgeführt bis eine der Abbruchkriterien

- Standardabweichung (STABW) ist kleiner 5 % oder
- Nur vier R-Werte vorhanden und  $STABW < 5.25\%$  oder
- Nur drei R-Werte vorhanden und  $STABW < 5.75\%$

zutrifft.

Die finalen Werte der Standard-Berechnung werden in *PRN\_standards\_fin.csv*, durch `write_std()` festgehalten.

Nach der manuellen oder automatischen Modifizierung der Standardwerte wird die Subroutine `calculate_erg()` (Seite 54) aufgerufen, in welcher die Endergebnisse berechnet werden. Diese werden letztlich in eine Gesamtergebnistabelle übertragen, da die Ergebnisse der Auswertung von mittleren und langen Messungen nebeneinander pro Probe dargestellt werden sollen, um die weitere Bearbeitung der Daten zu erleichtern.

*Parameter:*

- Tabelle der Datei *nuklide\_mittel.csv*
- Tabelle der Datei *probendaten.csv*
- Pfad des ausgewählten Auswertungsordner
- Tabelle für Gesamtmesswertausgabe
- Tabelle für Gesamtergebnisausgabe
- Probenreihenname

Zu niedrige Analyseparameter können zu der Fehlermeldung „One or more peaks were dropped due to Multiplet-deconvolution.“ führen. Diese bedeutet, dass entweder die Schwelle der Peaksignifikanz für die Peaksuche oder die Schwelle der Residuum-Suche bei der Peakflächenintegration zu tief angesetzt wurde und aus diesem Grund die Teilung des Multipletts in Einzelpeaks nicht korrekt durchgeführt werden konnte.

Nähere Ausführungen zur instrumentellen Neutronen-Aktivierungsanalyse und entsprechender Fehlerkorrektur sind in Kapitel 2.1.2. angeführt. Erstellte Ausgabedateien, wie Logfile, Messwerttabellen, Standardwertetabellen, und Ergebnistabellen sind in Kapitel 5.3 mehrmals dargestellt. Der Programmcode ist vollständig für die Funktionen `evaluation_mittel()` und `evaluation_lang()` in Anhang C angeführt.

#### 4.5.2.2. Erste Berechnung

Im ersten Schritt der Berechnung wird mittels Gleichung (3) die Zählrate pro Gramm Proben einwaage  $\left[\frac{1}{s \cdot g}\right]$  berechnet.

Die Peakfläche wird vom korrigierten LFC - Spektrum extrahiert, die Halbwertszeit wird in der jeweiligen Datei *nuklide\_mittel.csv* oder *nuklide\_lang.csv* bereitgestellt. Die Abklingzeit  $t_{ab}$  wird berechnet, indem die Startzeit der ersten Probe als Referenz-Nullpunkt definiert wird und von der Startzeit der Messung der aktuellen Probe subtrahiert wird. Beide Messstartzeiten werden aus den Reportdateien ausgelesen.

Die Messdauer  $t_{mess}$  nimmt normalerweise nur zwei Werte - 1800 s bei der mittleren und 10000s bei der langen Messung - an, wird aber ebenfalls aus der Reportdatei ausgelesen.

`calculate_1(double(,), string(,), string(,), integer, integer, integer, integer)`

In einer Schleife werden die Zählraten pro Gramm aller Nuklide für eine Probe, nach (3), berechnet, wenn der Wert der Peakfläche aus der Reportdatei nicht Null ist.

*Parameter:* Werttabelle der Reportdatei  
Tabelle der Datei nuklide\_mittel.csv oder nuklide\_lang.csv  
Tabelle der Datei probendaten.csv  
Nuklidanzahl  
Abklingzeit  
Messdauer  
Probennummer

### 4.5.2.3. Standards

Im zweiten - wichtigsten - Schritt der Auswertung werden die R-Werte der Standards manuell vom Benutzer (Kapitel 6.1.7.4) oder automatisch überprüft und bearbeitet. Dies richtet sich vor allem nach der gewünschten Genauigkeit und Schnelligkeit der Auswertung. Die automatische Auswertung ist nahezu instantan abgeschlossen und somit weitaus schneller als manuelles Kontrollieren und Entfernen der Werte aller Nuklide durch den Benutzer.

Der Vorteil der manuellen Auswertung besteht insbesondere darin, dass in manchen Fällen - in denen jeweils zwei R-Werte ähnlich sind - eine etwas höhere Genauigkeit erreicht werden kann. Wenn eine höhere Genauigkeit als eine Standardabweichung von 5% erwünscht ist, ist dies ebenfalls nur manuell zu bewerkstelligen, da bei automatischer Auswertung ab Standardabweichungen kleiner 5 % keine weiteren Werte entfernt werden.

Die Erstellung der R-Wertetabelle der Standards sowie die Berechnungen des Mittelwerts der R-Werte, der Standardabweichung und der Standardfehler des Mittelwerts, werden in `calculate_std()` durchgeführt. Berechnet werden die Eintragungen nach den Ausführungen in Kapitel 2.1.2.5.

`calculate_std(double(,), string(,), integer, List(of string))`

Variablen werden definiert und die Standard-Wertetabelle dimensioniert sowie beschriftet. Daraufhin wird eine Doppelschleife initiiert, die Position des aktuell betreffenden Standards definiert und für jede Energielinie

- die Bezeichnungen der Zelle in die Tabelle eingetragen,
- die R-Werte der Standards nach (4),
- der Mittelwert der R-Werte nach (5),
- die Standardabweichung der R-Werte vom Mittelwert nach (6),
- den Standardfehler des Mittelwerts (absolut) nach (7),
- den Standardfehler des Mittelwerts (relativ) und
- das Quadrat des Standardfehler des Mittelwerts (relativ)

berechnet.

*Parameter:* Ergebnistabelle der ersten Berechnung  
Tabelle der Datei nuklide\_mittel.csv oder nuklide\_lang.csv  
Nuklidanzahl  
Liste der Bezeichnungen und Positionen der Standards in Probenliste

Wenn durch die manuelle/automatische Bearbeitung R-Werte entfernt wurden, werden mittels Aufruf von `recalculate_std()` der Mittelwert und alle Fehlerangaben der Standardtabelle, neu berechnet.

`recalculate_std(string(,), List(of string))`

Die Zellen der neu zu berechnenden Werte werden in der Standard-Ergebnistabelle geleert. Anschließend wird eine Doppelschleife initiiert, welche für jede Energielinie, ident wie in `calculate_std()`,

- die Bezeichnungen der Zelle in die Tabelle einträgt und
- die R-Werte der Standards (4),
- der Mittelwert der R-Werte (5),
- die Standardabweichung der R-Werte vom Mittelwert (6),
- den Standardfehler des Mittelwerts (absolut) (7),
- den Standardfehler des Mittelwerts (relativ) und
- das Quadrat des Standardfehlers des Mittelwerts (relativ)

berechnet.

*Parameter:*           Ergebnistabelle der Standard-Berechnung  
Liste der Bezeichnungen und Positionen der Standards in  
Probenliste

#### 4.5.2.4. Ergebnis

Im letzten Schritt der Auswertung wird aus den Vorberechnungen sowie den erstellten und modifizierten Referenzwerten der Standards die Konzentration  $K_{\text{Probe}}$  der Nuklide in den Proben mit (1) sowie dazugehörige Fehler mit (10) berechnet.

`calculate_erg(string(,), double(,), double(,), string(,), string(,), integer)`

Zuerst werden Bezeichnungen der Ergebnistabelle festgelegt und anschließend eine Doppelschleife gestartet, welche die Konzentrationen jedes Nuklids in jeder Probe berechnet. In derselben Schleife wird der Fehler der Peakfläche mit dem Fehler des Mittelwerts nach (10) addiert und durch Multiplikation mit der Konzentration als absoluter Fehler zur weiteren Bearbeitung zur Verfügung gestellt.

*Parameter:*           Endergebnistabelle  
Messwertdaten  
Ergebnistabelle der ersten Berechnung  
Ergebnistabelle der Standard-Berechnung  
Tabelle der Datei probendaten.csv  
Anzahl der Standards

#### 4.5.2.5. Einlesefunktionen

Einlesefunktionen sind Routinen, mit welchen Daten aus bereitgestellten Ressourcendateien importiert werden:

`get_probendaten(string)`

Importiert die gesamte Tabelle aus probendaten.csv (Seite 48). Mittels der Klasse Streamreader wird ein Objekt instanziiert und die Ressource definiert. Innerhalb einer Schleife wird das gesamte Dokument eingelesen und durch mehrere Schritte in einzelne Tabellenwerte getrennt, damit diese in abrufbaren, variablen Arrays gespeichert werden können. Die Überschriften der Tabelle in probendaten.csv werden nicht eingelesen, die Rückgabetable ist aber mit drei Spalten fi-

xiert initialisiert.

*Parameter:* Pfad der Ressource probendaten.csv

#### `get_nuklide_data(string)`

Importiert die gesamten Tabellen aus nuklide\_mittel.csv bzw. nuklide\_lang.csv (Seite 48). Mittels der Klasse Streamreader wird ein Objekt instanziiert und die Ressource definiert. Innerhalb einer Schleife wird das gesamte Dokument eingelesen und durch mehrere Schritte in einzelne Tabellenwerte getrennt, damit diese in abrufbaren, variablen Arrays gespeichert werden können. Durch variable Dimensionierung der Rückgabetablelle unterliegen den Ressourcendateien kaum Einschränkungen.

*Parameter:* Pfad der Ressource nuklide\_mittel.csv oder nuklide\_lang.csv

Reportdateien werden mit der Funktion `get_rpt_table()` eingelesen. Es ist darauf zu achten, dass das Format der Reportdateien der Funktion angepasst sein muss und aus diesem Grund entsprechende Reportvorlagen bei der Analyse verwendet werden müssen. Weitere Informationen werden in Kapitel 6.1.1.6 gegeben. Zum besseren Verständnis wird in Abbildung 21 ein Auszug einer Reportdatei - erstellt mit dem Template `a_heinz.tpl` - gezeigt.

#### `get_rpt_table(string)`

Importiert die Wertetabelle der übergebenen Reportdatei (Kapitel 6.1.1.5). Mittels der Klasse Streamreader wird ein Objekt instanziiert und die Ressource definiert. Innerhalb einer Schleife wird das Dokument bis zum Terminator „Errors ted“ eingelesen. Durch mehrere Schritte werden die Dezimalzeichen der Werte, wenn nötig durch Beistriche ersetzt. Der Beginn jeder eingelesenen Zeile wird Zeichen für Zeichen gestutzt bis ein Terminator „F“, „M“, „m“, die Zahlen 1-10 oder „or more peaks“ erreicht ist. Es werden einige Terminatoren benötigt, da unterschiedliche Parameter oder Analysemethoden zu verschiedenen Ausgabeformaten oder fehlerhaft analysierte Peaks zu Fehlermeldungen (Kapitel 6.1.5.4) führen können. Die Fälle unterschiedlicher Terminatoren werden in der Folge für die Trennung der einzelnen Werte einer Zeile berücksichtigt. Fehlermeldungen führen unter anderem dazu, dass eingelesene Werte des entsprechenden Peaks gleich Null gesetzt werden. Zur Fehlervermeidung wird das Format der Reportdatei - sieben Spalten - überprüft. Abschließend wird die Datentabelle der Reportdatei der Quellfunktion zurückgegeben.

*Parameter:* Pfad der einzulesenden Reportdatei

Aufgrund der verschiedenen Methoden - Standardanalyse, Spezialanalyse - und unterschiedlichen Analyseparametern - Analysebereich, Einzellinien-Fit, zur Erzeugung von Reportdateien, weisen diese, trotz Reportvorlagen, einige Unterschiede auf. In Abbildung 22 ist ein Ausschnitt einer Reportdatei, erstellt mit der Methode „Spezialanalyse“, ersichtlich.

```

*****
A n a l y s e n b l a t t
*****

Filename: C:\auswertung\18A\lang\18A_ESB_LANG_0001_LFC.CNF
ID: Q0318LFC/SApeaksLAN/

Acquisition Start Time: 12.03.18 08:03:25
Live Time:                10000 seconds
Real Time:                10000 seconds

Peak Analysis Performed:  29.03.2018 11:57:52
Analysis from channel:    309
Analysis to channel:      384

Peak No.   Energy [keV]   Net Peak Area[cts]   Bkgnd [cts]   FWHM [keV]   Counts/sec   Error [%]
↑ M 1      121.61           355556             694682        1.500         3.556E+001    0.65
m 2        123.18           77944              693076        1.503         7.794E+000    1.16
m 3        130.35           31541              736883        1.511         3.154E+000    1.92
m 4        132.91          100936             734748        1.514         1.009E+001    0.83
m 5        135.96           48115              732256        1.518         4.811E+000    1.32
m 6        142.48           60513              727420        1.525         6.051E+000    1.26

```

Abb.21: beispielhafter Auszug einer Reportdatei; Screenshot des Verfassers

```

↑ F 1      1188.95           19969             25602         2.142         1.997E+000    3.25
↑ M 1      1212.88             4561              16611         2.138         4.561E-001    2.59
m 2        1221.33           31331             17317         2.141         3.133E+000    0.64
m 3        1230.97           13551             15794         2.145         1.355E+000    1.09
↑ M 1      1400.59             1400              6171          2.218         1.400E-001    5.45
m 2        1408.14           63762             7482          2.221         6.376E+000    0.82
↑ 1        1691.40           20912             5352          2.361         2.091E+000    0.94
↑
Errors quoted at 1.00 sigma

```

Abb.22: Ausschnitt einer Reportdatei, erstellt mit Spezialanalyse; Screenshot des Verfassers

Die Pfeile bedeuten, dass mehrere Berichte unterschiedlicher Analyseschritte einer Reportdatei angefügt wurden. Der Parameter „Fitte Einzellinien“ führt zur Peakbezeichnung „F“ und bedeutet, dass Einzelpeaks gefittet wurden. Ist dieser Parameter nicht ausgewählt, wird bei Einzelpeaks keine Peakbezeichnung eingefügt. Multipletts in Analysenbereichen haben Zeichen „M“ - für den ersten Peak eines Multipletts und „m“ für die weiteren Peaks desselben Multipletts zur Folge. Die nachfolgenden Zahlen zeigen die Anzahl der gefundenen Peaks pro Analysebereich an. Um eine einheitliche Tabelle einlesen zu können, müssen die Zeilen angeglichen und die Diskrepanzen der Zeilenanfänge entfernt werden.

`get_acqstarttime(string)`

Extrahiert aus der aktuellen Reportdatei die Messstartzeit / das Messstartdatum, ersichtlich in Zeile 9 von Abbildung 21.

Dazu wird die Reportdatei eingelesen und danach in mehreren Schritten die entsprechende Zeile in einzelne Werte und die Datums-/ Uhrzeitangabe in einzelne Teile für Jahr, Monat, Tag, Uhrzeit geteilt. Die Zeitangabe wird im gewünschtem Format erstellt und der aufrufenden Funktion zurückgegeben.

*Parameter:* Pfad der aktuellen Reportdatei

`get_acqduraction(string)`

Extrahiert aus der aktuellen Reportdatei die Messdauer, ersichtlich in Zeile 10

von Abbildung 21.

Dazu wird die Reportdatei eingelesen und die entsprechende Zeile in einzelne Werte getrennt.

Die Messzeit wird der aufrufenden Funktion zurückgegeben.

*Parameter:* Pfad der Reportdatei des ersten LFC-Spektrums

#### 4.5.2.6. Ausgabefunktionen

Ausgabedateien sind durchwegs CSV-Dateien mit englischen Formatspezifikationen, also „," als Trennzeichen und „.“ als Dezimalzeichen. Der Grund dafür ist wiederum die erleichterte Weiterbearbeitung der Daten, da Datenimportfunktionen von Programmen üblicherweise für englische Spracheinstellungen standardisiert sind.

Die Funktion `write_str_tbl()` wird aufgerufen um die initialen sowie die finalen Standardwerte in die Dateien `PRN_standards_init.csv` und `PRN_standards_fin.csv` auszugeben sowie die Dateien `PRN_messwerte.csv` und `PRN_auswertung.csv` mittels den zugehörigen Tabellen zu erstellen. `write_logfile()` wird nur einmal, zur Erstellung einer Übersicht der Referenzen aller ausgewerteten Spektren in `PRN_logfile.csv`, aus `auswerten()`, aufgerufen. Diese Funktion wird für erhöhte Übersichtlichkeit aus der Stammfunktion `auswerten()` ausgegliedert. Die Abkürzung PRN steht für Probenreihenname. Dieser spezifiziert die Dateinamen der Ausgabedateien. Die Funktion `write_line()` dient zur Ausgabe einzelner Zeilen.

`write_str_tbl(string, string(,))`

Gibt unter Berücksichtigung des englischen Ausgabeformats die übergebene Tabelle mittels der Streamwriter-Klasse in eine vordefinierte Ausgabedatei (.csv) aus.

*Parameter:* Pfad der Ausgabedatei  
auszugebende Tabelle

`write_logfile(string)`

Überschriften der Datei werden erstellt und anschließend werden zuerst die Spektren der mittleren Messung geöffnet und Dateinamen sowie Referenzen in die Datei `PRN_logfile.csv` übertragen. Im zweiten Schritt wird die gleiche Prozedur für die Spektren der langen Messung durchgeführt.

*Parameter:* Pfad des Stammordners

`write_line(string,string)`

Mittels der Streamwriter-Klasse wird ein Objekt initialisiert, wodurch die Ausgabedatei definiert wird. Der übergebene String wird in die Zieldatei übertragen. Hauptsächlich dient diese Funktion zur Erstellung von Überschriften und Ausgabe von Leerzeilen für eine bessere Übersichtlichkeit.

*Parameter:* Pfad der Zieldatei  
auszugebende Zeichenkette

#### 4.5.2.7. Fehlerbehandlung in der Auswertungsklasse

Die Fehlerbehandlung der Auswertung gründet sich hauptsächlich in der Kontrolle von Ordnerstruktur sowie Ein- und Ausgabedateien durch die Funktionen `check_dir()` und `check_files()`, mit der Kopplung der Rückgabewerte an die Fehlervariable `error_num`.

`check_dir(string)`

Dient zur Überprüfung aller Unterordner des Stammordners, gemäß der Ordnerstruktur in Abbildung 20 - im Speziellen der Ordner „auswertung“, „mittel“ und „lang“ sowie den beiden darin befindlichen „reports“ Unterordnern. Wird ein angeführter Ordner vermisst, wird der Quellfunktion ein Fehlerwert zurückgegeben und eine entsprechende Fehlermeldung ausgegeben.

*Parameter:* Pfad des Stammordners der Auswertung

`check_files(string, string)`

In dieser Funktion werden zwei unterschiedliche Probleme gelöst. Einerseits werden zu erstellende Ausgabedateien auf deren Existenz überprüft, damit diese nicht überschrieben werden können und andererseits wird die Existenz von benötigten Ressourcendateien, welche Daten und Parameter für die Auswertung enthalten, gecheckt. Insbesondere werden alle Reportdateien der mittleren und langen Analyse sowie die Dateien `probandaten.csv`, `nuklide_mittel.csv` und `nuklide_lang.csv` auf Vollständigkeit überprüft und bei negativem Ergebnis ein Fehlercode zurückgegeben.

Ein Fehler wird ebenfalls ausgelöst, wenn eine der Dateien `PRN_logfile.csv`, `PRN_messwerte.csv`, `PRN_standards_init.csv`, `PRN_standards_fin.csv` oder `PRN_auswertung.csv` bereits vorhanden ist. PRN steht für Probenreihenname und spezifiziert den Dateinamen.

*Parameter:* Pfad des Stammordners der Auswertung  
Probenreihenname

Zusätzlich sind in der gesamten Auswertung fehlervermeidende Abfragen implementiert, wie zum Beispiel die Kontrollen, ob die Anzahl der Reportdateien pro Ordner gerade und doppelt so hoch ist wie die Probenanzahl. Wäre dies nicht der Fall, ist entweder die Analyse nicht vollständig durchgeführt oder die Datei `probandaten.csv` nicht korrekt erstellt worden. Folgend wird eine entsprechende Fehlermeldung ausgegeben und die Auswertung abgebrochen.

Die Auswertung wird ebenfalls mit entsprechender Fehlermeldung von unvollständigen Auswertungsdateien abgebrochen, wenn eine Standard-Form bei manueller Überprüfung der Standardwerte unautorisiert beendet oder ohne positiven Rückgabewert abgebrochen wird.

#### 4.5.3. Tochterklasse von Auswertung.vb

Die Klasse „Auswertung“ hat nur eine Tochterklasse „Standards“, welche in den Funktionen `evaluation_mittel()` und `evaluation_lang()` instanziiert und aufgerufen wird, wenn durch den Benutzer die manuelle Bearbeitung der Standardwerte innerhalb der Auswertung gewählt wurde. Die Anpassung der Benutzeroberfläche bzw. die Eintragung der Werte und die Bezeichnungen der Labels finden in der Quellfunktion statt.

Der Aufruf der Einstiegsfunktion führt zur Anzeige der Standard-Form, beispielhaft für ein Nuklid und fünf Standards abgebildet in Abbildung 23. Bis zu zehn Standards können in dieser Form dargestellt werden, wobei die Anzeige der Labels mit der Anzahl der Standards variiert. Zusätzlich zu den R-Werten der Standards werden Mittelwert und Standardabweichung angezeigt. Nach diesen Werten hat der Benutzer zu entscheiden ob R-Werte, mittels der Entfernen Buttons [x], gelöscht werden sollen. Die Form beinhaltet drei Buttons - Speichern, Abbruch und Zurück, welche durch die Funktionen Bsave\_Click, Bcancel\_Click und Bundo\_Click abgewickelt werden. Der Zurückbutton ist nur aktiviert, wenn bereits ein Standard - R-Wert entfernt wurde.

Die Klasse beinhaltet drei öffentliche Variablen

- **Deleted\_items** beinhaltet ein Array aus zehn Nullen und dient als Markerarray, welche Werte vom Benutzer entfernt wurden.
- **Delete\_stack** beinhaltet eine Liste von Integerwerten, welche dazu dient eine Liste der gelöschten Werte zu erstellen, damit diese nach dem Stapelprinzip reversiv wiederhergestellt werden können.
- **All\_val** beinhaltet die R-Werte der Standards, damit für alle Funktionen der Klasse Zugriff auf die R-Werte besteht.

**Abb.23:** beispielhafter Aufruf der Standardform für ein Nuklid und fünf Standards; Screenshot des Verfassers

`showform(integer(), string())`

Die R-Werte der Standards werden einer öffentlichen Variable übergeben und anschließend die Form mittels ShowDialog() aufgerufen. Positive bzw. negative Rückgabewerte durch Betätigen der Buttons Speichern und Abbruch führen zu entsprechenden Rückgabewerten der Funktion und Änderung des Markerarrays.

*Parameter:* Markerarray, welches gelöschte Werte anzeigt  
Array aller R-Werte

Die Funktionen Bsave\_Click und Bcancel\_Click sorgen für positive und negative Rückgabewerte der ShowDialog()-Funktion und schließen die Instanz der Form.

Für jeden „Löschen-Button“ [x] jedes Standards der Form existiert eine Funktion - B\_di\_Click() mit i = 1 bis 10, folgend beispielhaft für i=1 beschrieben:

B\_d1\_Click()

- Dem Löschstapel wird der Wert „1“ zugefügt.
- Das entsprechende Label wird geleert.
- Im Markerarray wird an der Stelle 0 der Wert „1“ für True gesetzt.
- Die Funktion recalc() wird aufgerufen.
- Der Zurückbutton wird aktiviert.

recalc()

Gemäß den Berechnungen in calculate\_std() (Seite 53) werden die Anzahl der R-Werte gezählt und anhand dieser neuen Werte Mittelwert und Standardabweichung nach (5) und (6) neu berechnet und in die entsprechenden Labels übertragen.

Bundo\_Click()

Das oberste Element des Löschstapels wird identifiziert und das zugehörige Label mit dem ursprünglichen Wert - gespeichert in all\_val - gefüllt. Das Element des Markerarrays wird wieder auf „0“ für FALSCH gesetzt und der Eintrag am Löschstapel entfernt. Ist der Löschstapel leer wird der Rückgängig-Button deaktiviert. Abschließend werden wiederum Mittelwert und Standardabweichung, wegen geänderten R-Werten, mittels recalc() neu berechnet.

## **5. ÜBERPRÜFUNG DER ERGEBNISSE**

Offensichtliche Verbesserungen ergeben sich aus der Modernisierung der Messcomputer. Durch die identischen Spezifikationen der beiden Messcomputer wird die zukünftige Modifizierung und Durchführung von Updates der PC's erleichtert. Die Geschwindigkeit der Rechner ist um ein Vielfaches gestiegen und durch die Erneuerung der Betriebssysteme ist die Absturzgefahr der Messsysteme stark gesunken.

Um Verbesserungen bzw. mögliche Verschlechterungen von bisherigem zu neuem Mess-, Analyse- und Auswertungssystem herauszuarbeiten, werden die Veränderungen der einzelnen Bereiche gegenübergestellt.

### **5.1. ÜBERPRÜFUNG DER MESSUNG**

Da der Messaufbau im Hinblick auf den Detektor, Probengeometrie und Korrektursystem und Messmethode nicht verändert wurde, ist kein Unterschied der aufgezeichneten Spektren zu ermitteln. Die CAM-Dateien unterscheiden sich nur insofern von den bisherigen, dass bisher ungenützte Beschreibungsparameter für Zwecke der Reproduzierbarkeit mit Referenzen gefüllt werden (Kapitel 4.2.1)

Erkennbare Unterschiede ergeben sich nur in der Anwendung durch den Benutzer sowie der Fehlerbehandlung bezüglich Probleme bei der Anwendung des Programms. Benutzereingaben werden überprüft (Kapitel 4.2.2.3. & Kapitel 4.3.2.1.) und bestehende gleichnamige Spektren-Dateien werden bei Verwendung des neuen Systems nicht überschrieben. Eine Vorwahl, betreffend mittlerer oder langer Messung, kann vom Benutzer getroffen werden, wodurch Messdauer sowie Dateinamenspezifikationen automatisch definiert werden.

Bezüglich der Geschwindigkeit der Messung sind keine bemerkenswerten Unterschiede festzuhalten, da die Auswirkungen der effizienteren Rechenleistung der Mess-PC's im Vergleich der langen Messzeiten irrelevant sind.

### **5.2. ÜBERPRÜFUNG DER ANALYSE**

Schwer vergleichbare Verbesserungen sind die Einführung der vorher nicht vorhandenen Analysemethoden wie Kalibrierung, Interactive-Peak-Fit sowie die speziell angepasste Analysemethode. Die Kalibrierung und der IPF sind und waren über das Programm Genie2000 durchzuführen. Die Verbesserung besteht in der Möglichkeit mehrere Spektren, nachträglich und gleichzeitig zu kalibrieren oder mit dem IPF zu bearbeiten. Mit dem Programm DUAL\_LFC können Kalibrierungen und Bearbeitungen effizienter durchgeführt werden. Die einzelnen Teile der speziellen Analysemethode sind ebenfalls in Genie2000 zugänglich. Damit diese Analysemöglichkeiten effizient genug verwendet werden können, müssen diese innerhalb einer Routine automatisiert durchgeführt werden. Dies ist mit DUAL\_LFC möglich.

Für Anwendungen, welche nicht mit der Analysefunktion „Spezialanalyse“ bearbeitet werden sollen oder können, ist die bisherige Analysemethode durch Auswahl einer Analysesequenz in der Routine „Standardanalyse“ verwirklicht. Die Standardanalyse unterscheidet sich zur vorherigen Analyse wiederum nur durch das Auftreten der Referenzen in den Reportdateien und durch die Wahl der Reportvorlage (Kapitel 6.1.1.6.).

Durch die spezifizierte Analysemethode, durch Anpassung der Analyseparameter in peaks.txt, können Verbesserungen bezüglich der Suche nach Peaks und der Integration der Flächen, gegenüber einer Standardanalyse, herausgearbeitet werden. Dies erfolgt durch Vergleich von Messwerttabellen, welche mit unterschiedlichen Inputs, wie Reportdateien aus verschiedenen Analysemethoden, erzeugt wurden.

Durch die spezifische Anpassung der Parameter der Spezialanalyse (Tabelle 4) an die einzelnen Energielinien der Nuklide werden signifikant mehr Peaks gefunden, welche durch die Analyse mit Standardparametern nicht gefunden wurden. Dies liegt daran, dass durch die Berücksichtigung niedriger Signifikanzschwellen auch Peaks mit schwacher Intensität gefunden werden können. In den Ausschnitten der Abbildungen 24 und 25 ist an der Menge der gelben Zellen ersichtlich, wie viele Peaks bei Verwendung der Standardanalyse vs. Verwendung der Spezialanalyse nicht registriert werden können. Diese Abbildungen dienen nur zur Veranschaulichung der geringeren Häufigkeit der gelb hinterlegten Zellen. Den Werten ist demnach keine Aufmerksamkeit zu widmen.

Wenn Proben ein Nuklid einer entsprechenden Energielinie nicht oder in zu schwacher Konzentration enthalten, tritt in den zugehörigen Spektren kein oder ein im Hintergrund verschwindender Peak auf.

**Abb.24:** Darstellung der Verbesserung der Peaksuche bei der mittleren Messung der Probenreihe 17G durch Parameteranpassung; Es soll die starke Verringerung der Häufigkeit der gelben Zellen durch die Anpassung gezeigt werden. Die Messwerte haben in dieser Abbildung keine Wichtigkeit; Links - Standardanalyse, Rechts - Spezialanalyse; Screenshot des Autors



Eher untypisch sind die vermehrt auftretenden Abweichungen der Energielinie von Arsen. Diese Abweichungen wurden stichprobenartig mittels Interactive-Peak-Fit manuell untersucht. Dabei wurde festgestellt, dass die Peaks mit der speziellen Analysemethode korrekt gefittet werden, wenn ein genügend großer Bereich zur Peaksuche gewählt wird, damit die Höhe des Hintergrundes richtig berechnet wird. Die Abweichungen von Lutetium entstehen durch die Schwierigkeit alle Peaks des Multipletts zu finden. Dazu wurden die Energien Samarium, Lutetium und Neptunium genauer betrachtet. Die Samarium-Linie befindet sich in einem Multiplett von sechs Energielinien, wobei der sechste Peak sehr klein und diesen zu finden nicht relevant ist. Die Lutetium-Linie befindet sich in einem Duplett und die Neptunium-Linie in einem Triplet. Mit Standardparametern werden bei allen Proben im Sm-Multiplett nur drei Peaks, im Lu-Duplett nur ein Peak und im Np-Triplett nur zwei Peaks gefunden. Ziel ist es, durch die Anpassung der Analyseparameter, möglichst alle Peaks der Multipletts bei allen Proben zu finden. Dies musste bisher mühevoll von Hand nachgeführt werden.

Durch die Anpassung der Parameter werden in der Probenreihe 17G im Samarium-Multiplett bei allen Proben mindestens fünf Peaks gefunden. Im Lutetium-Duplett wird bei 10 von 50 Proben der Probenreihe der zweite Peak nicht gefunden und im Neptunium-Triplett wird bei 4 von 50 Proben der dritte Peak nicht gefunden. Daraus lässt sich eine starke Verbesserung zur alten Analysemethode erkennen. Dennoch bleibt die Energielinie von Lutetium problematisch. Dieser muss weiterhin Aufmerksamkeit gewidmet werden.

	Sm-153	Lu-177	Np-239 (U)	Ar-76	W-187	Na-24	K-42	La-140	Na-24																		
CFA	2E+06	0,22	0,3447	68031	4,26	3,1005	72908	2,59	2,4747	678706	0,34	0,4775264	15134	5,46	0,7929	215475	0,25	0,3496	19011	1,5	1,1467	666285	0,1	0,2172	172290	0,3	0,1681
LSS	1E+06	0,32	0,4336	35614	1,93	0,7341	37331	3,48	0,7518	100745	0,93	2,6502556	5624	10,53	4,463	1361671	0,1	0,3348	25559	1,29	0,8451	330230	0,7	1,1004	748248	0,1	0,1006
GBW	2E+06	0,26	0,0777	43096	6,95	2,2262	43173	3,57	5,4838	2357	14,7	0	3194	18,33	2,4421	2915094	0,07	0,4404	46540	1,07	0,8681	675676	1	1,1202	1609663	0,1	0,3046
SO-1	1E+06	0,33	0,1535	19259	5,01	1,945	12950	13,81	3,2947	10515	9,64	16,034237	3026	39,44	6,312	3279929	0,07	0,4659	28009	1,78	1,4638	476174	0,2	0,3912	1809376	0,1	0,0762
IMS	1E+06	0,29	0,2218	29709	6,85	1,3724	52281	0,77	0,4834	197202	0,52	1,1683451	12395	4,66	1,0246	791197	0,71	0,133	14952	1,81	0,1404	431116	0,2	1,2252	435522	0,2	0,1097
SATS	681099	0,52	0,2539	38487	8,43	2,9179	37677	4,54	0,9975	11694	4,6	0	3529	11,24	0	4344603	0,1	0,5333	21678	2,45	2,6202	203426	0,2	0,4237	2402189	0,1	0,0764
Bann	924813	0,33	0,9242	27922	6,88	0,5908	45513	2,52	0,1938	171734	0,68	1,1088913	11448	2,92	15,418	266591	0,47	0,1999	10544	1,91	0,1992	359522	0,2	0,1165	146450	0,3	0,0252
L651-02	462022	0,62	0,1542	28092	4,72	16,415	5547	14,06	3,1891	31621	3,62	0	2294	25,23	3030401	0,07	0,4157	6986	5,68	4,0366	97389	0,6	0,1581	1676043	0,1	0,282	
L651-02	770825	0,35	0,3628	27693	6,66	3,0733	13271	6,96	2,3304	29168	3	4,1312397	2285	10,28	28,359	1169493	0,11	0,3044	16788	1,72	0,2025	245216	0,7	0,2696	646741	0,1	0,0847
L654-01	370488	0,66	0,1443	21055	5,45	4,3992	5753	12,45	0,2114	18361	5,71	0	3533	20,58	2423318	0,02	0,3562	5260	6,81	1,73	75448	0,6	0,4984	1340586	0,6	0,1511	
L654-01	439201	0,57	0,1062	23604	4,5	5,8722	4853	11,47	5,1958	28991	3,85	0	3509	9,57	1489177	0,09	0,265	7631	4,03	3,1641	109746	1,6	1,7094	821729	0,1	0,1375	
L654-01	855503	0,43	0,3472	20523	7,74	0,1377	24260	5,03	0,2901	21810	1,15	6,96382329	2157	11	24,71	1186438	0,1	0,2697	14164	1,99	0,6848	168709	0,6	0,5453	649659	0,1	0,7258
L654-01	648424	0,41	0,2726	20180	3,86	0,7424	17706	5,67	0,8738	17523	4,22	4,2548337	3932	5,68	1,6781	736201	0,13	0,2194	16755	1,51	0,4476	231382	0,3	0,1872	404377	0,2	0,0564
L654-1C	597252	0,4	0,0006	22165	7,14	2,7136	23982	1,17	0,0023	23828	1,03	3,9652006	3121	23,75	36,622	681363	0,23	0,0527	15003	1,52	0,8665	192006	0,3	0,1363	374901	0,2	0,1184
L654-1I	629388	0,39	0,2144	22640	7,12	1,3055	17604	4,61	0,265	13807	1,54	9,2706598	2952	6,59	4,4038	471821	0,17	0,2064	16127	1,25	0,1488	211998	0,4	0,9708	258665	0,2	0,0305
L658-01	659874	0,38	0,6309	23974	8,76	5,4322	18450	4,92	0,446	22772	3,18	6,2364543	2868	6,73	1,6288	706246	0,13	0,1877	14785	1,56	0,2367	207520	0,3	0,8529	387983	0,2	0,0452
L659-01	621880	0,39	0,4257	21341	7,18	3,9544	17496	1,43	0,0025	15402	3,74	10,85777	3315	6,39	26,886	415563	0,34	0,1175	13800	1,51	0,2246	197092	0,3	0,035	227128	0,2	0,0332
L659-01	371052	0,34	0,1346	23726	5,89	0,7019	23242	4,25	0,4635	44619	0,63	4,6499510	5276	18,34	8,3207	647522	0,24	0,1995	18834	1,33	0,0531	306776	0,2	0,1803	267979	0,2	0,0566
L660-01	340027	0,45	0,0004	11475	12,4	0,3449	11610	5,2	1,0376	34047	1,6	0,0080813	1045	21,83	2,9665	131179	0,31	0,1456	3111	3,47	0,5464	96203	0,3	0,1019	72656	0,4	0,0454
L665-01	497721	0,47	0,081	18543	11,6	2,7058	14614	6,4	0,4744	32271	2,47	0	407	22,54	1436247	0,09	0,2346	7391	3,63	0,1353	127114	0,4	0,4044	796216	0,1	0,1324	
L665-01	215240	0,84	0,784	17144	12,7	6,3255	3549	12,52	1,0244	15093	5,71	0	0	0	572496	0,15	0,1907	2115	7,83	0,9929	46092	0,6	0,2539	316172	0,2	0,0915	
L667-01	504110	0,48	0,7187	21872	9	1,961	17257	5,95	0,2692	36340	0,77	1,6784525	2199	23,89	22,874	1122768	0,1	0,2364	9557	2,54	0,8546	127108	0,4	0,2179	616555	0,1	0,084
L668-01	465454	0,45	0,3902	18097	11,3	2,2313	12166	2,13	3,9295	35703	0,74	3,4646948	2127	45,22	40,056	780765	0,13	0,1598	10189	2,2	1,1581	124392	0,3	0,0079	426962	0,2	0,0223
L669-01	201364	0,91	0,3433	15240	11	1,9254	2747	7,83	10,269	4651	4,44	0	0	0	599435	0,14	0,1443	1930	8,15	4,6632	38212	0,7	0,3289	330250	0,2	0,0618	
L670-01	218386	0,81	0,4128	18053	10,9	1,0394	3092	7,76	5,7422	5698	3,51	0	1793	39,68	737952	0,13	0,1568	1747	10,4	0,3434	40640	0,7	1,1294	404840	0,2	0,0381	
L670-01	230790	0,86	0,0041	16522	5,5	3,2913	3970	6,82	2,0348	6035	7,75	0	2306	21,06	700664	0,13	0,2015	2902	7,43	2,1709	51388	0,6	0,4807	385399	0,2	0,048	
L673-01	628706	0,37	0,0043	27563	2,93	17,994	16602	4,95	0,3316	12011	4,85	4,2960619	3000	27,19	22,9	573106	0,15	0,1778	10762	1,97	0,0372	194737	0,3	0,1448	136248	0,2	0,0372
L673-1I	580056	0,39	0,4232	23124	7,55	0,6735	16391	2,1	6,1605	40828	0,65	1,9933095	2349	7,15	54,278	382334	0,18	0,2132	11196	1,68	0,2769	184614	0,3	1,4992	209801	0,2	0,0771
L673-1I	572873	0,4	0,8595	22294	6,83	1,3051	16440	4,99	0,2871	15744	1,28	8,3224085	3074	5,93	6,311	430140	0,17	0,2134	9997	1,96	1,4804	178548	0,3	0,1016	238804	0,2	0,085
L673-01	612275	0,37	0,466	23516	7,26	1,2821	17763	6,43	0,3971	93108	1,08	0,5531211	3423	26,45	15,746	421433	0,33	0,2776	9777	1,96	0,3704	189777	0,3	0,0316	231328	0,2	0,012
L673-01	197346	1,02	0,0193	15587	6,48	6,5879	2265	19,06	0,5276	29722	2,9	0,0033645	2953	13,3	1045459	0,11	0,2244	2144	9,68	5,1306	37900	0,8	0,2325	576539	0,2	0,0872	
L673-01	491266	0,46	0,0003	21581	8,11	0,3055	18793	5,91	1,3792	28426	2,36	2,2690495	2208	34,5	25,906	1000266	0,11	0,1777	6706	3,69	2,3889	129027	0,3	0,2868	549333	0,2	0,1034
L673-01	245447	0,74	0,083	16575	11,2	2,0884	3166	7,34	0,2956	8526	2,54	0	2408	35,7	975280	0,11	0,1863	3019	6,84	5,2004	44569	0,7	0,2926	537089	0,2	0,0472	
L673-01	320575	0,67	0,249	24076	4,07	16,276	4042	11,23	1,1193	28709	3,89	0	3104	30,11	932270	0,03	0,151	2908	6,53	5,095	63291	0,6	0,2092	510002	0,2	0,1271	
L673-01	502711	0,46	0,14	19137	9,76	5,0628	17614	5,73	0,1176	43769	1,49	1,5855971	2531	43,45	960777	0,11	0,207	6675	3,59	1,8577	135959	0,3	1,0466	530227	0,6	0,096	
L674-01	308231	0,69	0,4809	19657	5,8	8,6396	2283	11,64	6,8576	32304	3,14	0	0	0	864705	0,12	0,1986	3260	6,83	10429	72609	0,5	0,1694	477796	0,2	0,1051	

**Abb.26:** Auszug des Vergleichs der Messwerte der alten und neuen Analysemethode, der Spektren der mittleren Messung der Probenreihe 17G; Gezeigt wird in dieser Abbildung die Seltenheit von auftretenden Abweichungen zwischen alter Analysemethode mit manueller Korrektur und neuer speziellen Analysemethode. Auftretende Abweichungen sind großteils auf bestimmte Nuklide beschränkt. ; Screenshot des Verfassers

Bei Vergleich der Messwerte der langen Messung tritt ein ähnliches Bild auf. Die Messwerte der neuen Analysemethode sind wesentlich exakter. Bei problematischen Energielinien kann wieder eine deutliche Verbesserung der Peaksuche, aus den unverkennbar häufiger gefundenen Peaks bei angepassten Analyseparametern. Zudem sind häufiger auftretende Abweichungen bei Terbium ersichtlich, welche großteils nur knapp über

5 % liegen und nach manueller Stichproben-Untersuchung vorerst ebenfalls als annehmbar betrachtet werden.

Da die Anpassung der Analyseparameter ein langwieriger Prozess ist, soll die Qualität der Messwerte mit jeder weiteren Probenreihe kontinuierlich durch sukzessives Anpassen der Parameter verbessert werden.

Ein Nachteil der speziellen Analyse ist die Möglichkeit, dass zu niedrige Analyseparameter definiert werden könnten. Dies führt zu Problemen bei der Aufspaltung der Multipletts in Einzelpeaks und in weiterer Folge zu Fehlermeldungen in der Reportdatei und somit zu fehlenden Werten der ausgewerteten Energielinie. Sind Parameter zu hoch angesetzt, kann dies dazu führen, dass Peaks nicht gefunden werden. In Kapitel 4.4.2.3 werden die Parameter näher erläutert.

### 5.3. ÜBERPRÜFUNG DER AUSWERTUNG

Als nächstes werden die initial berechneten R-Werte der Standards überprüft. Dazu wurden die R-Werte der mittleren Messung der Probenreihe 17G der alten Auswertung und der neuen Auswertung, exportiert aus der Ausgabedatei *PRN\_standards\_init.csv*, verglichen. In Tabelle 7 sind alle Nuklide dargestellt, welche Abweichungen aufweisen.

								>5 %
								2 - 5%
								< 2 %
NEU	ALT	absolut	relativ [%]	NEU	ALT	absolut	relativ [%]	
Lu-177				Np-239 (U)				
0,26699649	0,27527459	0,0082781	0,03100454	0,03914089	0,04010949	0,00096861	0,02474667	
0,29398073	0,29613874	0,00215801	0,00734065	0,03853962	0,03824987	0,00028976	0,00751842	
0,2861386	0,2925086	0,00637	0,02226193	0,04016858	0,04237137	0,00220278	0,05483844	
0,25682867	0,26182402	0,00499535	0,01945013	0,03175591	0,0306779	0,00107801	0,03394672	
				0,02741516	0,02754769	0,00013253	0,00483407	
As-76				W-187				
0,02360071	0,02371341	0,0001127	0,00477526	0,01280634	0,01290789	0,00010154	0,00792917	
0,02186655	0,02244607	0,00057952	0,02650256	0,0177398	0,01853153	0,00079173	0,04463016	
0,01634854	0,01634854	3,4694E-17	2,1222E-15	0,01332985	0,01365537	0,00032553	0,02442079	
0,02262595	0,02625385	0,0036279	0,16034237					
0,02454725	0,02483405	0,0002868	0,01168345	0,0113304	0,0112143	0,00011609	0,01024607	

**Tab.7:** Auszug des Vergleichs der Standard-Referenzwerte der alten und neuen Auswertung, der mittleren Messung von Probenreihe 17G; Farblich hinterlegt sind die relativen Abweichungen der Referenzwerte von der alten und neuen Auswertung; Lila hinterlegte Zellen bedeuten, dass diese Werte bei manueller bisheriger Auswertung entfernt wurden. ; Screenshot des Verfassers

Die erste Spalte jedes Nuklids beinhaltet den R-Wert der neuen Auswertung, die zweite den R-Wert der alten Auswertung, die dritte und vierte Spalte zeigen die absolute und relative Abweichung der R-Werte. An den hinterlegten Farben erkennt man eine Abstufung der relativen Abweichungen, wobei die lila hinterlegten Werte bedeuten, dass die betreffenden R-Werte während der alten Auswertung manuell entfernt wurden. Die Abweichungen zwischen zwei und fünf Prozent bei den bekannt problematischen Nukliden Lutetium, Neptunium und Wolfram sind akzeptabel, da in der alten Auswertung der Großteil dieser Werte manuell ent-

fernt wurde. Näher zu betrachten hingegen sind die Werte mit einer Abweichung größer 5 %, welche in der alten Auswertung nicht entfernt worden sind. Der dritte Standard weist bei Neptunium eine Abweichung von 5.5 % auf, bei Vergleich des alten mit dem neuen R-Wert, wird offenbar, dass der neue R-Wert näher am Mittelwert aller R-Werte liegt und dementsprechend eher verwendet werden kann, als der alte Wert. Gleiches zeigt sich auch beim Vergleich des R-Werts des vierten Standards bei Arsen, welcher eine Abweichung von 16 % zeigt. Bei allen anderen Nukliden und Standards beträgt die Abweichung unter 2 %.

<b>Np-239 (U)</b>			
	NEU	ALT	Mittelwert
<b>GBW</b>	0,040168584	0,042371366	0,036205538

**Tab.8:** Vergleich alter und neuer R-Wert des Standards GBW bei Neptunium-239

<b>As-76</b>			
	NEU	ALT	Mittelwert
<b>SO-1</b>	0,022625948	0,026253846	0,021797799

**Tab.9:** Vergleich alter und neuer R-Wert des Standards SO-1 bei Arsen-76

Anschließend werden durch den Benutzer oder automatisch die R-Werte der Standards bearbeitet. Dabei werden R-Werte entfernt, welche zu weit vom Mittelwert entfernt sind, wenn die Standardabweichung der Werte über 5 % liegt. Zielsetzung ist in beiden Vorgehensweisen eine Standardabweichung kleiner 5% mit möglichst großer Anzahl an R-Werten zu erreichen. Die Kriterien der automatischen Abweichung sind für die Routine gestaffelt auf Seite 51 ersichtlich.

Für den Vergleich der bearbeiteten Standards wird die Ausgabedatei *PRN\_standards\_fin.csv* herangezogen. Die Werte der Standardtabelle der neuen Auswertung werden mit denen der alten Auswertung verglichen. Es handelt sich wiederum um die Probenreihe 17G.

In den Tabellen 10 und 11 sind die bearbeiteten Standardwerte der mittleren und langen Messung der Probenreihe 17G dargestellt. Anhand der grün hinterlegten Werte ist offensichtlich, dass die Zielsetzung einer Standardabweichung kleiner 5 % bei der mittleren Messung durchwegs erreicht und bei der langen Messung ebenfalls bei allen Nukliden, mit Ausnahme von Zirkonium, mit guter Qualität erreicht wurde. In der Zeile „ALT“ unter der Wertetabelle kann die Größe der Standardabweichung der alten Auswertung und eine klare Verbesserung der Qualität zur neuen Auswertung festgestellt werden.

Die automatische Bearbeitung wird gegenüber der manuellen Bearbeitung der R-Werte ebenfalls überprüft. In guter Näherung stimmen die Werte überein. Wie bereits erwähnt, kann eine etwas geringere Standardabweichung anhand manueller Bearbeitung erreicht werden, da weitere R-Werte entfernt werden können, auch wenn die Standardabweichung bereits unter 5 % liegt.

CFA	0,58186695	0,26699649	0,03914089	0,02360071	0,01280634	0,00074697	4,65E-06	0,0334899	0,00040794
LSS	0,6202421	0,29398073	0,03853962			0,00072536	4,94E-06	0,03526336	0,00039859
GBW	0,65609878	0,2861386	0,04016858		0,01332985	0,00071757	4,91E-06	0,03694835	0,00039623
SO-1	0,64683874			0,02262595		0,00072973	4,81E-06	0,03730488	0,00040256
IMS	0,6546775			0,02454725		0,00068521	4,39E-06		0,00037718
Mittelwert	0,63194481	0,28237194	0,03928303	0,0235913	0,01306809	0,00072097	4,74E-06	0,03575162	0,0003965
Standardabweichung	0,04983208	0,04915805	0,02096912	0,04072201	0,0283265	0,0314966	0,0472938	0,04897825	0,02944866
Fehler des Mittelwerts(FdM)	0,01408326	0,00801412	0,00047558	0,00055465	0,00026175	1,02E-05	1,00E-07	0,00087553	5,22E-06
FdM %	0,02228558	0,02838141	0,01210653	0,02351087	0,02002986	0,01408571	0,02115043	0,02448913	0,01316984
FdM %^2	0,00049665	0,0008055	0,00014657	0,00055276	0,0004012	0,00019841	0,00044734	0,00059972	0,00017344
<b>STABW ALT</b>	<b>0,04882211</b>	<b>0,05660512</b>	<b>0,05128817</b>	<b>0,06667699</b>	<b>0,09932231</b>	<b>0,03062325</b>	<b>0,04466305</b>	<b>0,06176445</b>	<b>0,02938303</b>

**Tab.10:** Vergleich der bearbeiteten „alten“ und „neuen“ Standardabweichungen der Standard-Referenzwerte der Probenreihe 17G - MITTEL

	Ce-141	Yb-169	Lu-177	Pa-233 (Th)	Cr-51	Hf-181	Ba-131	Sr-85	Nd-147	Zr-95	Cs-134
CFA	0,00192888	0,00768853	0,02530823	0,01782057	0,00100665	0,0119238	4,06E-05	3,54E-05	9,93E-05		0,01125465
LSS	0,00190287	0,00744539	0,02681826	0,01731394	0,0010851	0,01310113	3,81E-05		0,00010208	2,92E-05	0,01192208
GBW	0,00197732	0,00790973	0,02779562	0,01799525		0,01225757	3,78E-05	3,62E-05		2,56E-05	0,01124658
SO-1		0,00735787	0,02526022	0,01692636	0,00102246	0,01278017	4,01E-05	3,65E-05	0,00011104		0,01178468
IMS	0,00183262			0,0164547			4,04E-05		0,00010443		0,01087508
Mittelwert	0,00191042	0,00760038	0,02629558	0,01730217	0,00103807	0,01251567	3,94E-05	3,60E-05	0,00010423	2,74E-05	0,01141661
Standardabweichung	0,03159025	0,03278792	0,04693769	0,03665243	0,03996748	0,04201767	0,03399466	0,01534945	0,04794729	0,09455497	0,03765861
Fehler des Mittelwerts(FdM)	3,02E-05	0,0001246	0,00061713	0,00028361	2,40E-05	0,00026294	5,99E-07	3,19E-07	2,50E-06	1,83E-06	0,00019227
FdM %	0,01579512	0,01639396	0,02346884	0,01639147	0,02307524	0,02100884	0,01520288	0,00886201	0,02397364	0,06686046	0,01684144
FdM %^2	0,00024949	0,00026876	0,00055079	0,00026868	0,00053247	0,00044137	0,00023113	7,85E-05	0,00057474	0,00447032	0,00028363
<b>STABW ALT</b>	<b>0,05447248</b>	<b>0,03246764</b>	<b>0,04332932</b>	<b>0,03774936</b>	<b>0,03959762</b>	<b>0,02981169</b>	<b>0,03845061</b>	<b>0,04592129</b>	<b>0,05302346</b>	<b>0,10592924</b>	<b>0,04236564</b>
	Co-58 (Ni)	Tb-160	Rb-86	Fe-59	Zn-65	Sc-46	Co-60	Ta-182	Ta-182	Eu-152	Sb-124
CFA	0,00021559	0,02192765	0,00024756	9,94E-06	0,0002212	0,09771795	0,00611602	0,00872004	0,01360645	0,01198806	
LSS	0,00021221	0,02113957	0,00024442	1,03E-05	0,00021833	0,10137981	0,00606231	0,00896957	0,01391358	0,012048	0,00307376
GBW	0,00022954	0,02140795		9,74E-06	0,00022947	0,10400643				0,01142835	
SO-1	0,00021036		0,00023383	9,82E-06	0,00021632	0,09858046	0,0060615	0,00895289	0,01411113	0,01268643	
IMS	0,0002154		0,00024228		0,00020059		0,00561365				0,00329568
Mittelwert	0,00021662	0,02149172	0,00024202	9,95E-06	0,00021718	0,10042116	0,00596337	0,00888083	0,01387705	0,01203771	0,00318472
Standardabweichung	0,03484979	0,01864276	0,02427136	0,02585579	0,04853218	0,02843933	0,03933014	0,01570793	0,01832642	0,04275654	0,04927314
Fehler des Mittelwerts(FdM)	3,38E-06	0,00023132	2,94E-06	1,29E-07	4,71E-06	0,00142796	0,00011727	8,05E-05	0,00014683	0,00025735	0,00011096
FdM %	0,0155853	0,0107634	0,01213568	0,0129279	0,02170425	0,01421966	0,01966507	0,00906898	0,01058076	0,02137827	0,03484137
FdM %^2	0,0002429	0,00011585	0,00014727	0,00016713	0,00047107	0,0002022	0,00038671	8,22E-05	0,00011195	0,00045703	0,00121392
<b>STABW ALT</b>	<b>0,03251808</b>	<b>0,02047988</b>	<b>0,05920098</b>	<b>0,05001026</b>	<b>0,04991408</b>	<b>0,02850914</b>	<b>0,03779033</b>	<b>0,05794137</b>	<b>0,01832642</b>	<b>0,04749151</b>	<b>0,37894202</b>

**Tab. 11:** Vergleich der bearbeiteten „alten“ und „neuen“ Standardabweichungen der Standard-Referenzwerte der Probenreihe 17G - LANGE

Als wichtigstes Werkzeug zur Kontrolle der Funktionstüchtigkeit des Programms ist der Vergleich der Ergebnisse der bisherigen Auswertung mit den Ergebnissen der neuen Auswertung - verfügbar durch die Ausgabedatei *PRN\_auswertung.csv*, welche durch das Programm *DUAL\_LFC* erzeugt wird.

In Abbildung 27 ist der Ausschnitt des Vergleichs der Auswertungsergebnisse dargestellt, in welchem die Abweichungen am größten sind. Die Abweichungen der übrigen Nuklide - Samarium, Natrium, Kalium und Lanthan - betragen durchwegs weniger als 3 %. Ersichtlich ist in den drei Spalten eines Nuklids das Ergebnis von *DUAL\_LFC*, der Fehler von *DUAL\_LFC* und die relative Abweichung der neuen Peakfläche zur alten Peakfläche - mit Farben von Grün über Gelb bis Rot hinterlegt. Bei grünem Hintergrund ist die Abweichung klein, bei rotem groß. An den blau hinterlegten Zellen erkennt man, wieviele Peaks durch die alte Auswertung nicht, mit der neuen Methode aber dennoch gefunden werden.

Lu-177		Np-239 (U)		As-76		W-187				
1,154582723	0,895892892	3,449145147	0,258198874	0,828774849	186,2548219	0,824223151	2,588935557	5,788766893	0,898749376	0,825771976
0,468588265	0,891539186	1,688938664	3,551493564	0,822298971	3,128722261	16,19715717	0,825418552	0,8382828	1,827661557	0,827624918
0,678997371	0,898736491	2,55377771	4,398882882	0,825448489	2,361628488	0,457979528	0,845889177	2,869787282	1,225882142	0,892882079
0,241957497	0,891455777	2,399891946	1,382942552	0,893804929	5,788472263	1,191888885	0,893814547	12,52528887	1,116149189	0,897519645
0,263728822	0,898874117	1,293933554	7,258881924	0,814925288	7,194952886	47,4358878	0,814547419	1,888818894	6,848522382	0,891857141
0,678295956	0,898248444	3,268838532	4,829454155	0,824585884	1,418165286	2,538216488	0,891824148	2,363797282	1,283788455	0,841688291
0,454284584	0,898458558	0,245978884	5,465294985	0,895642898	2,197619546	36,4882937	0,824915887	1,887884639	4,855282592	0,891886649
0,446788288	0,895742287	18,14567481	0,653691959	0,894482856	0,726166255	6,562924588	0,892451882	2,363797282	0,752581119	0,847811686
0,466399119	0,89868626	3,455884624	1,663188355	0,828827155	0,142829124	4,484812289	0,822882148	1,845883344	1,891882762	0,848522891
0,368251495	0,897499119	4,814239886	0,754848855	0,897984118	2,188582781	4,194758896	0,89522674	2,363797282	0	0
0,356494622	0,895499189	5,253915181	0,612825537	0,89566574	2,844915881	6,582528861	0,896622862	2,363797282	1,621521289	0,893636882
0,368249888	0,899748866	0,194288624	3,28787367	0,825486799	2,188882722	5,288887789	0,825841185	3,63668849	1,858553884	0,841481795
0,348287865	0,894518871	1,788291882	2,261167642	0,826718828	1,593885459	4,891628261	0,891221277	1,164994966	1,871464857	0,84931823
0,375882994	0,898888837	3,164162864	3,882352264	0,816294998	2,381881889	5,68888574	0,82687891	0,788888836	1,587497886	0,854672893
0,488881616	0,89882521	1,64246847	2,424878848	0,824648894	2,645497962	3,522882889	0,82585142	6,892464227	1,238888814	0,858681198
0,424888922	0,84188622	5,28253624	2,58876452	0,825278898	3,545888856	5,768884954	0,82588875	3,888416884	1,488164884	0,8958788
0,368885225	0,898821464	3,88825691	2,325116149	0,891818884	2,88288619	3,879474822	0,898482872	7,572297245	1,688751195	0,853988784
0,45616875	0,89393929	1,62424658	3,894242567	0,829878885	1,323548889	11,26758842	0,824814586	1,742474782	2,782878667	0,851614582
0,289425844	0,846382872	0,788557887	1,64241459	0,825818153	1,374885547	3,288785838	0,82637644	2,372287486	0,685888841	0,872458824
0,324888282	0,844887917	3,847444882	1,387355644	0,828488293	1,323748871	0,451987968	0,828288117	2,363797282	0,453635939	0,848188779
0,29593244	0,845887862	0,61419729	0,888249495	0,898724894	3,886786511	3,916918868	0,89522668	2,363797282	0	0
0,38719297	0,84128789	1,355192969	2,397193722	0,827294885	2,826454825	3,844812982	0,82935188	1,395828884	1,827849561	0,842884898
0,322772885	0,848828291	2,619393782	1,799478299	0,898865714	6,222518829	18,844821442	0,825351582	0,898291888	1,729197486	0,871652783
0,282281914	0,848658638	4,89979512	0,88228714	0,898484828	7,897448882	6,88854885	0,81571927	2,363797282	0	0
0,328595958	0,84557482	0,71865915	0,447828854	0,898374523	7,992885594	1,648482938	0,898862919	2,363797282	0	0
0,325852488	0,89881788	4,946398879	0,497427959	0,828785476	4,79882829	1,522286984	0,89498764	2,363797282	1,289528783	0,852154592
0,48871468	0,89148714	17,27488291	2,357488888	0,829299998	2,74856796	3,49882828	0,892146455	1,285888874	1,78888888	0,857788874
0,41873896	0,89589288	0,49935915	2,386739286	0,818882886	3,62588882	12,14775452	0,824854854	1,126541458	1,449488819	0,861458863
0,376842382	0,89851812	0,377182142	2,249788185	0,825488174	2,188578842	4,429947992	0,82881529	5,537188249	1,798788192	0,832548819
0,421491988	0,89194151	1,187528847	2,588723227	0,828899889	2,774464876	28,8556676	0,82578294	2,427882382	2,149815887	0,858888266
0,228628977	0,898142972	0,27788694	0,264849377	0,853885855	1,871824784	7,262891489	0,829884822	2,88472488	1,581516357	0,844891691
0,318247686	0,848828899	1,355192969	2,265141285	0,827158928	1,848888811	7,1212519579	0,82884986	0,791988485	0,289169799	0,844484168
0,38152514	0,848984844	1,62828688	0,475818814	0,826782823	2,88881888	2,793222814	0,82888888	2,363797282	1,154528885	0,864884897
0,35471794	0,84842812	19,88881959	0,498791989	0,895615555	0,59474288	2,44748837	0,898824889	2,363797282	1,681188888	0,85392875
0,381865497	0,842297959	4,470752826	2,292884824	0,82684884	1,789399948	12,26442958	0,827935989	1,425198188	1,482428848	0,874428782
0,3684511	0,848688879	0,35558892	0,947888821	0,896288888	4,887197672	18,7288819	0,829488879	2,363797282	0	0
0,384681462	0,841934591	4,66164155	2,851911987	0,826697883	4,88248887	6,429748896	0,82553556	2,363797282	0,378885884	0,842791693
0,521679888	0,89787211	0,562888888	3,197175894	0,822216524	2,728888794	0,398889395	0,828441889	0,516828884	1,454611992	0,853635888
0,489288886	0,89418949	16,7286754	0,762951945	0,898522894	2,35288788	3,886814836	0,829795889	2,363797282	2,381578959	0,857198886
0,389358641	0,898821667	0,148259851	2,61524729	0,829578828	1,217878856	3,995441937	0,828891497	0,293949394	2,8528879	0,119788445
0,468876678	0,89193577	1,199198877	2,636885822	0,829814896	2,381491884	7,299881884	0,82566452	0,488878944	2,866286955	0,897888885
0,394643954	0,84667999	15,27788897	0,24658885	0,84789188	0,888888886	3,956758415	0,82542612	2,363797282	0	0
0,21919882	0,84668918	0,54839371	0,32937419	0,845588886	2,891994	2,849588827	0,86725871	2,363797282	0	0
0,528915598	0,89829858	3,43632984	0,882958879	0,82582286	0,19979894	5,182749158	0,895893564	2,363797282	0,155888468	0,875888647
0,329474999	0,84648192	2,54765896	2,814882749	0,828955119	2,88862569	7,624978866	0,82788874	4,966488677	1,881978888	0,897881422
0,365797988	0,898817268	0,888825195	0,389354194	0,848758723	0,61142699	2,496821811	0,89787812	2,363797282	1,64248744	0,842888884
0	0	0	0	0,824628883	0,573954122	7,851144936	0,828493978	2,363797282	0	0
0,399288867	0,842994914	1,589188469	0,848267548	0,829718995	0,76559882	6,192821772	0,82722671	2,363797282	0,374914886	0,878886274
0,429399747	0,846298557	0,767888853	2,587297984	0,817452893	1,879188995	3,884828858	0,828891696	2,361275888	0,841642694	0,729978891

**Abb.27:** Ausschnitt des Ergebnisvergleichs der alten und neuen Auswertungsmethoden anhand der Probenreihe 17G - MITTEL; Auch hier sind nur die farblichen Markierungen der Zellen und nicht die Messwerte an sich von Bedeutung. Pro Nuklid ist die farbig hinterlegte Spalte die relative Abweichung der neuen zur alten Peakfläche. Von niedriger Abweichung (grün < 5 %) bis hohe Abweichung (rot > 20 %); Blauer Hintergrund zeigen Peaks, welche nur mit neuer Analyseme-thode gefunden wurden; Screenshot des Verfassers

Da abgesehen von einzelnen Proben nur die problematischen Nuklide Abweichungen aufweisen, und die problematischen Nuklide ebenfalls eine sehr starke Verbesserung aufweisen, ist die neue Auswertungsmethode sehr zufriedenstellend.

Für den Vergleich der Ergebnisse der langen Messung werden in Abbildung 28 abermals nur auffallend interessante Nuklide dargestellt. Die übrigen Nuklide weisen abgesehen von vereinzelt größeren Abweichungen, nur Werte kleiner 3 % auf. Vereinzelt größere Abweichungen können beispielsweise dadurch erklärt werden, wenn der analysierte Peak Teil eines Multiplett ist. Mit der bisherigen Analyse-methode wurden eventuell nicht alle Peaks eines Multiplett gefunden. Sind derartige Peaks bei einzelnen Proben nun stärker ausgeprägt wie beim Großteil der Proben und werden diese mit der speziellen Analyse-methode registriert, ergeben sich dadurch einzelne stärkere Abweichungen.

Der Vergleich der Ergebnisse der langen Messung von Lutetium zeigt, dass das Nuklid bei der langen Messung weniger Abweichungen aufweist als bei der mittleren Messung. Für die weitere Bearbeitung sollten dementsprechend eher die Ergebnisse der langen Messung von Lutetium verwendet werden.



Eine weitere Vergleichsmöglichkeit bieten die zur Verfügung stehenden Standards. In der Probenreihe 17G sind sieben Standards - CFA, LSS, GBW, SO-1, IMS, SAT5 und Bonn, ersichtlich in den Nukliddateien „mittel“ und „lang“ - vorhanden. Da die Zusammensetzung der Standards bekannt ist und die jeweiligen Konzentrationen in den Nukliddateien angegeben sind, bietet es sich an, die Probenreihe so abzuändern als ob ein Standard als normale Probe gemessen würde. Diese Probenreihe anschließend auszuwerten und die Ergebnisse mit den bekannten Konzentrationen zu vergleichen, bietet eine weitere Kontrollmöglichkeit.

Durchgeführt wird diese Kontrolle für die Standards CFA, SO1, LSS, GBW und SAT5. Am Beispiel von CFA wird die Vorgehensweise veranschaulicht.

Es werden Probenreihen erstellt, in welchen die Standards als Proben definiert sind und als Standard entfernt werden, damit kein Zirkelschluss entsteht.

Name	Masse[mg]	Standard
CFA	118,09	0
LSS	150,05	LSS
GBW	125,57	GBW
SO-1	135,42	SO-1
IMS	104,78	IMS
SAT5	114,34	SAT5
Bonn	122,66	Bonn

**Tab.12:** Probenreihe zur Überprüfung der Konzentrationen des Standards CFA

Anschließend werden die Reportdateien ausgewertet und die Ergebnisse mit den bekannten Konzentrationen der Standards verglichen. Dabei wird festgestellt, dass die Elementkonzentrationen der fünf getesteten Standards teilweise stark abweichen. In folgender Tabelle sind die Nuklide dargestellt, deren Abweichung von der Nuklidkonzentration größer 5 % ist und größer ist als die mittlere Abweichung der Konzentrationen eines Nuklids der gesamten Probenreihe 17G. Kursiv dargestellt sind Nuklide die bekanntlich problematisch sind. Fett dargestellt sind Nuklide deren Verhältnis zwischen den Abweichungen der Konzentrationen und mittleren Abweichungen der Konzentrationen nicht vernachlässigbar groß sind.

Standard	Nuklide abweichender Konzentrationen
CFA	<i>Samarium, Lanthan, Antimon</i>
SO-1	<i>Lutetium, Neptunium, Lanthan, Cer, Hafnium, Zirkonium, Terbium, Tantal, Europium, Antimon</i>
LSS	<i>Lutetium, Arsen, Wolfram, Kalium, Chrom, Hafnium, Strontium, Cäsium, Eisen, Scandium, Antimon</i>
GBW	<i>Samarium, Lutetium, Arsen, Kalium, Lanthan, Ytterbium, Protactinium, Chrom, Kobalt, Rubidium, Zink, Scandium, Tantal, Antimon</i>
SAT5	<i>Samarium, Lutetium, Neptunium, Natrium, Tantal, Europium, Antimon</i>

**Tab.13:** Elemente mit abweichenden Konzentrationen - Standards werden nur als Probe ausgewertet

Diese Auswertungen wurden durch automatische Bearbeitung der Standards durchgeführt. Mittels manueller Bearbeitungen und Reduktion der Standardabweichung der R-Werte, könnten die Nuklidkonzentrationen exakter berechnet und die Häufigkeit von Abweichungen stark reduziert werden.

Die automatische Durchführung der Messung/Analyse/Auswertungs-Sequenz muss ebenfalls überprüft werden. Dazu wird die lange Messung der Probenreihe 18A manuell durchgeführt und eine automatische Sequenz der ersten zehn Proben derselben Probenreihe angehängt. Währenddessen können die Spektren der manuellen Messung bereits analysiert und ausgewertet werden. Anschließend werden die Ergebnisse der Ausgabedateien *PRN\_auswertung.csv* verglichen.

Dieser Vergleich ergibt, dass die Peakflächen durchwegs übereinstimmen bzw. größtenteils Abweichungen kleiner zwei Prozent zeigen. Eine geringe Abweichung ist akzeptabel, da Proben nicht gleichzeitig gemessen werden können und die Abklingzeit der Proben um beinahe 6 Tage länger ist und es sich außerdem um eine statistische Messmethode handelt.

Bei Konzentrationen mit großem Fehler zeigt sich, dass zwischen beiden Methoden teilweise große Abweichungen auftreten. Dies folgt direkt aus dem großen Fehler der Peakflächenberechnung. Der großen Übereinstimmung der meisten Werte zufolge liegen die Abweichungen nicht an der Funktion der automatischen Messung, sondern an der Tatsache, dass zweimal nacheinander gemessen und analysiert wird.

Die Durchführung der Sequenz kann somit mit Hinblick auf die Messung und Analyse als gleichwertig betrachtet werden. Die Auswertung kann klarerweise durch die manuelle Bearbeitung der Standards beeinflusst werden.

## 6. BEDIENUNGSANLEITUNG

Die Anleitung ist in zwei Bereiche gegliedert:

**Anwendung:** Beinhaltet die Verwendungsmöglichkeiten der einzelnen Methoden, deren Benutzereingaben und worauf bei Verwendung geachtet werden muss. Zusätzlich werden mögliche Fehlermeldungen und Fehlerquellen beschrieben.

**Implementierung:** Setup des Programms und Beschreibung der Ressourcen sowie Herangehensweise von Modifizierungen und Änderungen des Programms sowie mögliche Fehlerquellen

### 6.1. ANWENDUNG

Das Setup des Programms DUAL\_LFC aus Kapitel 6.2.1. muss abgeschlossen sein, um die Funktionstüchtigkeit der Anwendungen zu gewährleisten.

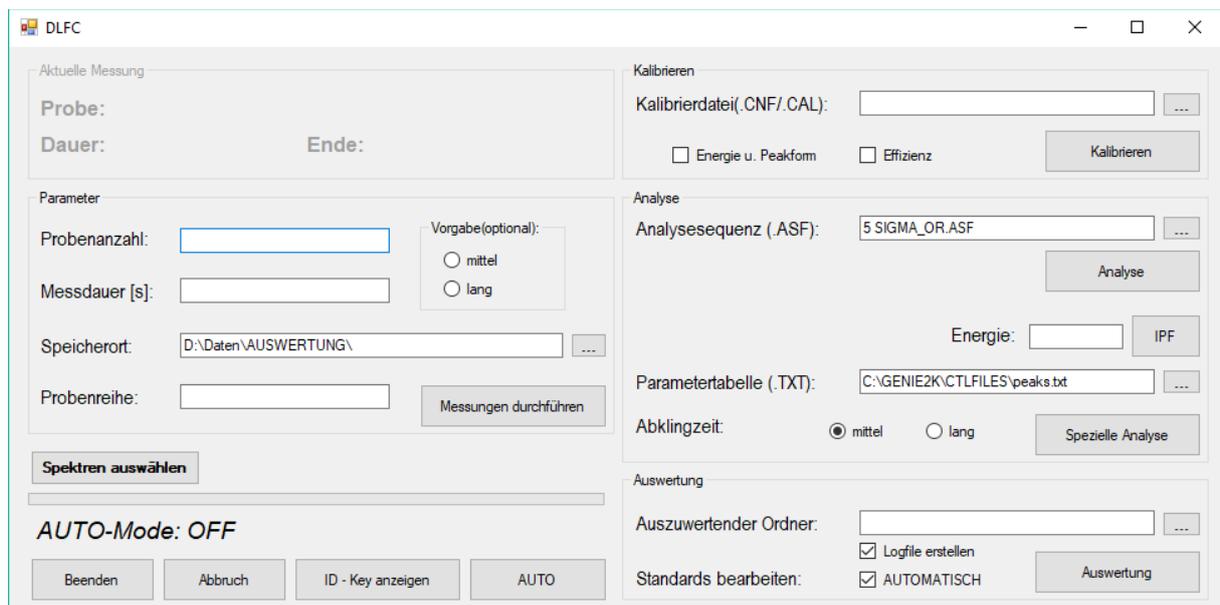


Abb.29: GUI des Programms DUAL\_LFC; Screenshot des Verfassers

#### 6.1.1. Ein- und Ausgabedateien

1. CAM-Dateien (.CNF)
2. Kalibrierdatei (.CAL)
3. Analysesequenzen (.ASF)
4. Parametertabelle (.TXT)
5. Reportdateien (.RPT)
6. Vorlagedateien (.TPL)

## 7. Textdateien (.CSV)

### 6.1.1.1. CAM-Dateien (.CNF)

CAM-Dateien werden automatisch im Zuge der Durchführung der Messung, durch Speicherung des Spektrums, erstellt. Sie beinhalten die zugehörigen Spektrendaten, sowie alle nötigen Parameter, beispielsweise Messstart, Messdauer, Kalibrierungsparameter und Referenzen. Nähere Beschreibungen sind im Genie-Manual [28] zu finden.

CAM-Dateien müssen für Analysemethoden ausgewählt werden und befinden sich standardgemäß in den Ordnern, entsprechend der Vorwahl „mittel/lang“ am ausgewählten Speicherort.

### 6.1.1.2. Kalibrierdateien (.CAL)

Enthalten den Block „Kalibrierung“ einer CAM-Datei, welcher alle nötigen Parameter zur Kalibrierung eines Spektrums beinhaltet. Sie können dazu verwendet werden mit der Methode „Kalibrierung“ (Kapitel 4.4.2.1.) auf mehrere Spektren die gleiche Kalibrierung zu übertragen. Nähere Beschreibungen des Kalibrier-Blocks und der enthaltenen Parameter sind im Genie-Manual [28] nachzulesen.

### 6.1.1.3. Analysesequenzen (.ASF)

Beschreiben vordefinierte Routinen, welche Schritt für Schritt abgearbeitet werden. Diese müssen für die Analysemethode „Standardanalyse“ (Kapitel 4.4.2.2.) ausgewählt werden um ausgewählte CAM-Dateien zu analysieren.

Vordefinierte Analysesequenzen befinden sich im Ordner C:\GENIE2k\CTLFILLES\. Fast ausschließlich verwendet wird die Analysesequenz „5 SIGMA\_OR.asf“.

Die Erstellung individueller Analysesequenzen wird im Kapitel 6.1.4.2. beschrieben.

### 6.1.1.4. Textdatei (.TXT)

Die einzige verwendete Textdatei mit dem Suffix „TXT“ dieses Programms ist peaks.txt, welche standardgemäß am Ort C:\GENIE2k\CTLFILLES\ gespeichert ist. Sie enthält alle Energien, welche analysiert und ausgewertet werden sollen sowie die zugehörigen Parameter, welche für spezifische Analysen von Spektren angepasst werden können (Tabelle 4).

Die Parametertabelle, deren Spalten, ebenso wie bei CSV-Dateien, mittels Komma separiert sind und als Dezimalzeichen ein Punkt verwendet wird, können dementsprechend in Excel als Textdatei importiert werden.

### **6.1.1.5. Reportdateien (.RPT)**

Reportdateien werden automatisch durch Bericht-Methoden, als Schritte von Analysesequenzen oder während der Durchführung der Spezialanalyse nach den definierten Vorlagendateien erstellt. Sie enthalten die analysierten Daten eines Spektrums, wie Energien mit dazugehörigen Peakflächen und Fehlern. Diese Daten werden für die Auswertungsroutine benötigt.

Das Format der Reportdateien ist durch die Vorlagen geregelt. Hauptsächlich werden zwei Vorlagen - a\_heinz.tpl und sanal.tpl - verwendet. Beide Vorlagen erstellen Reportdateien, deren Spalten durch Tabulatoren getrennt sind. Auszüge einer Reportdatei sind ersichtlich auf Seite 55 f.

Es ist darauf zu achten, dass die Verwendung des Dezimalzeichens und das Format der Datumsangaben von den Regioneneinstellungen des PC's abhängen. Für eine funktionierende Auswertung, muss aufgrund des geänderten Datumsformats, die Region „Österreich“ bzw. die Sprache „Deutsch“ definiert sein.

Der Ausgabeort der Reportdateien richtet sich nach der Auswahl der analysierten Spektren. Am Ort der CAM-Dateien wird ein Ordner „reports“ erstellt, in welchem die Berichte abgelegt werden.

### **6.1.1.6. Vorlagendateien (.TPL)**

Dienen als Vorlage zur automatischen Erstellung der Reportdateien. Vorlagendateien sind in den entsprechenden Berichtsfunktionen zu definieren und meistverwendete Vorlagendateien sind a\_heinz.tpl und sanal.tpl, welche beide am Ort C:\GENIE2k\CTLFILLES\ zu finden sind. A\_heinz.tpl wird üblicherweise für Standardauswertungen nach der Analysesequenz „5 SIGMA\_OR“ verwendet und sanal.tpl wurde eigens für die Spezialanalyse erstellt.

In Vorlagendateien können beispielsweise Abschnitte definiert werden, welche separat für Reportdateien aufgerufen und bestehenden Reportdateien angehängt werden können. Zur Bearbeitung oder Erstellung von Vorlagendateien wird auf das Genie-Manual [28] verwiesen.

### **6.1.1.7. Textdateien (.CSV)**

Enthalten Datentabellen zur Ein- bzw. Ausgabe von benötigten Parametern und Ergebnissen. Grundsätzlich werden alle Ein und Ausgabedateien mit Textinhalt als Textdatei im Dateiformat „comma-separated values“ (CSV) nach englischsprachigen Regioneneinstellungen erstellt, da die weitere Verwendung der Ausgabedaten mit diesem Format erleichtert wird. Dies bedeutet unter anderem, dass als Dezimalzeichen ein Punkt, als Trennzeichen ein Komma verwendet wird.

Diese Vorgabe sollte in allen Klassen des Programms realisiert sein, insbesondere aber der Klasse Auswertung und deren Eingabedateien:

- nuklide\_mittel.csv
- nuklide\_lang.csv

- probendaten.csv

Die Eingabedateien nuklide\_mittel.csv sowie nuklide\_lang.csv befinden sich am Ort C:\GENIE2k\CTLFILLES\ und enthalten alle auszuwertenden Nuklide, deren Halbwertszeiten, Energien und Suchintervalle. Zusätzlich enthalten die Dokumente die aktuell verfügbaren Standards sowie deren Nuklidkonzentrationen. Die Nukliddatei „mittel“ ist ersichtlich auf Seite 49.

Durch variable Dimensionierung der Rückgabetable unterliegen den Ressourcendateien keine Erweiterungseinschränkungen

In der - für jede Probenreihe unterschiedlichen - Datei probendaten.csv sind die Bezeichnungen der Proben und deren eingewogene Masse in Milligramm anzuführen. In der dritten Spalte ist definiert, ob die Probe als Standard ausgewertet werden soll oder nicht („0“). Soll eine Probe als Standard ausgewertet werden, muss der korrekte Name des Standards - ersichtlich in nuklide\_mittel.csv - eingetragen werden. Die Datei probendaten.csv muss im Ordner „auswertung“ des auszuwertenden **Stammordners** z.B. C:\Dokumente\18G\auswertung\probendaten.csv vorhanden sein. Die Überschriften der Tabelle in probendaten.csv werden nicht eingelesen.

Die Ausgabedateien der Auswertung *PRN\_logfile.csv*, *PRN\_messwerte.csv*, *PRN\_standards\_init.csv*, *PRN\_standards\_fin.csv* und *PRN\_auswertung.csv* werden am gleichen Ort wie probendaten.csv ausgegeben.

## 6.1.2. Messung

Eine Messung kann nur an vollständigen Messsystemen, beschrieben in Kapitel 3.2.3, durchgeführt werden.

Ausgabedateien sind CAM-Dateien (.CNF) der gesamten Probenreihe. Pro Probe werden drei CAM-Dateien - für korrigiertes (LFC), unkorrigiertes (UNC) und kombiniertes (DUAL) Spektrum ausgegeben.

### 6.1.2.1. Kalibrierung

Vor Starten der Messung sollte eine, an die Probenreihe angepasste, Kalibrierung am Detektor gespeichert sein, da diese Kalibrierung in die CAM-Dateien der Proben der gesamten Probenreihe übertragen wird. Dementsprechend muss die erste Probe manuell mit dem Probenwechsler in den Detektor eingebracht werden.

Mit dem Programm Genie2000 muss die eingebrachte Probe einige Minuten gemessen werden, damit anschließend eine Kalibrierung durchgeführt werden kann.

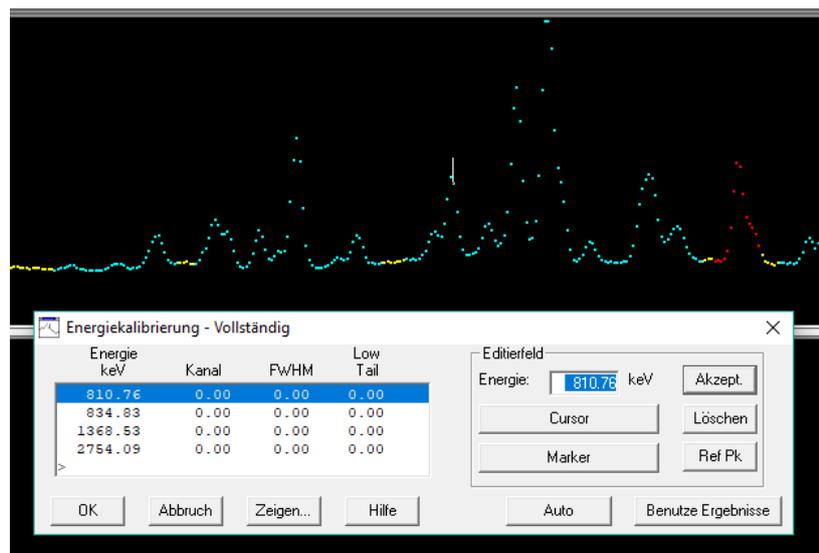
Die Kalibrierung dient dazu gemessene Peaks deren Energien zuzuweisen. Dabei werden die Kanäle des Multikanalanalysators, in welchen die Zählraten der Energien gespeichert sind, mit den bekannten Energien von Nukliden verknüpft. Damit diese Verknüpfung erstellt werden kann, müssen die Peaks im Spektrum markiert werden. Dazu muss eine Peaksuche

durchgeführt werden. Eine schnelle Möglichkeit alle Peaks des Spektrums zu markieren, ist *Analyse - Ausführen Sequenz - 5 Sigma ohne Residuum* durchzuführen.

Danach werden in *Kalibrierung - Energie & Peakform* einige passende Nuklide, welche sicher in der Probe vorhanden sind, ausgewählt und anschließend nacheinander alle Energielinien den Peaks im Spektrum zugeordnet, indem der Cursor an der Peakspitze positioniert und der Button „Cursor“, ersichtlich in Abbildung 30, betätigt wird.

Dies erzeugt eine lineare Beziehung zwischen den Kanälen des Spektrums und den Energien der Peaks. Zur Kontrolle muss mit dem Button „Zeigen“ die Linearität der Kalibrierung überprüft werden. Kalibrierpunkte, welche von der Geraden weit entfernt sind, sollten gelöscht werden. Ebenfalls muss die „Form“-Kalibrierung überprüft werden. Dazu kann mittels des Radiobutton „Form“ gewechselt werden. Dabei müssen ebenfalls von der Kurve stark abweichende Punkte entfernt werden. Die abgeschlossene Kalibrierung wird direkt am Detektor gespeichert.

Eine Kalibrierung dieser Art kann mittels *Kalibrieren - Sichern* gespeichert werden. Diese erzeugt eine Kalibrierdatei (.CAL), deren Verwendung in Kapitel 6.1.3 beschrieben wird. Weitere Möglichkeiten der Kalibrierung, deren Eigenschaften und Methoden sind im Genie Manual [26] näher definiert.



**Abb.30:** Darstellung der Kalibrierung nach Nuklidliste eines Spektrums; Screenshot des Verfassers

### 6.1.2.2. Benutzereingaben - Messung

Für die Messung stehen die Benutzereingaben der Gruppe Parameter - Probenanzahl, Messdauer, Speicherort, Probenreihe und Vorwahl „mittel/lang“ zur Verfügung. Die Vorwahl richtet sich nach der Abklingzeit der Proben nach der Aktivierung, also ob die Messung 5 oder 28 Tage nach der Aktivierung durchgeführt wurden.

Für die Eingaben Probenanzahl und Messdauer sind nur Zahlen erlaubt. Bei der Definition des Probenreihennamens werden Sonderzeichen unterdrückt.

Es ist darauf zu achten, dass bei manueller Eingabe eines Spektren-Speicherortes - im Gegensatz zur Auswahl mittels Browser-Dialog - es dem Benutzer obliegt, einen korrekten, existierenden Pfad zu wählen. Bei fehlenden Eingaben kann die Messung nicht gestartet werden.

Wird eine spezifische Messung - mittel oder lang - durchgeführt, sollte die Vorwahl zuerst getroffen werden, da die Auswahl für Probenzahl und Messdauer Default-Werte definiert und vorherige Eingaben überschreibt. Als Probenanzahl wird 50 eingetragen. Für mittlere Messungen ist eine Messdauer von 1800 Sekunden und für lange Messungen eine Messdauer von 10000 Sekunden vordefiniert.

### 6.1.2.3. Starten der Messung

Vor Start der Messung muss das Programm Genie2000 geschlossen werden.

Sind die Benutzereingaben vollständig, kann die Messung mittels „Messungen durchführen“ gestartet werden.

Im Bereich „Aktuelle Messung“ erscheinen die Angaben, welche Probennummer aktuell gemessen und wie lange diese noch gemessen wird. Diese Anzeigen aktualisieren sich regelmäßig. Die Anzeige „Ende“ zeigt das Ende der kompletten Messung aller Proben der Probenreihe an.

Während der Messung werden alle Teile der Benutzeroberfläche, außer den Buttons Abbrechen und Beenden, deaktiviert. Wird die Messung vorzeitig abgebrochen, müssen die Daten als nicht vollständig betrachtet werden.

### 6.1.2.4. Fehlerquellen und -meldungen der Messung

- Es wurde bereits eine Messung durchgeführt und im betreffenden Ordner sind gleichnamige CAM-Dateien vorhanden, wie die, die durch die bevorstehende Messung erstellt würden.
- Bei laufender Messung darf kein Programm, welches auf den Detektor Zugriff hat, geöffnet werden. Im speziellen muss das Programm Genie2000 geschlossen sein, da auch die dabei verwendeten Canberra-Klassen Fehler erzeugen können.
- Klarerweise darf trotz Trennung der Methoden in Klassen keine zweite Instanz des Programms DUAL\_LFC geöffnet bzw. verwendet werden, da infolgedessen Störungen im Programmablauf auftreten können.

"Fehlende Parameter!"

- ➔ Eine der Benutzereingaben Probenanzahl, Messdauer, Speicherort oder Probenname fehlt.

"Speicherort existiert nicht!"

- ➔ Der ausgewählte Pfad in „Speicherort“ ist nicht korrekt oder leer.

„Datei ist im Zielordner bereits vorhanden! Messabbruch.“  
“[Spektrumpfah]”

- ➔ Die Datei [Spektrumpfah] würde durch die bevorstehende Messung überschrieben werden. Der Dateiname muss geändert oder die existierenden Spektren entfernt/verschoben werden.

"Messung wurde abgebrochen!"

"Daten sind nicht vollständig - Spektren nicht verwenden!"

- ➔ Der Abbruch- oder Beenden Button wurde betätigt.

### 6.1.3. Kalibrierung

Soll eine bestehende Kalibrierung gleichzeitig auf mehrere Spektren übertragen werden, ist dafür die Komponente Kalibrierung zu verwenden.

Eingabedateien sind zu kalibrierende Spektren (.CNF)

#### 6.1.3.1. Benutzereingaben - Kalibrierung

Mit dem Button „Spektren auswählen“ sind alle Spektren auszuwählen, auf welche eine Kalibrierung übertragen werden soll. Der vollständige Pfad einer CAM-Datei (.CNF) oder einer gespeicherten Kalibrierung (.CAL) muss in „Kalibrierdatei“ enthalten sein. Diese Auswahl sollte, um Fehler zu vermeiden, mittels der Browser-Dialog-Funktion getroffen werden. Dabei werden alle angezeigten Dateien mit den Dateisuffixen „.CAL“ bzw. „.CNF“ gefiltert. Im Browser-Dialog muss der gewünschte Typ der Kalibrierdatei ausgewählt werden.

Durch Auswählen der Checkboxes „Energie u. Peakform“ und „Effizienz“ können die zu übertragenden Kalibrierparameter definiert werden. Genauere Beschreibungen der Parameter sind im Genie-Manual [28] zu finden.

Wurden alle Benutzereingaben gemacht, kann die Kalibrierung durchgeführt werden.

#### 6.1.3.2. Fehlermeldungen

"Keine Kalibrierung ausgewählt."

- ➔ Es wurden keine zu übertragenden Kalibrierungen - Checkboxes - ausgewählt.

"Korrekte Kalibrierdatei (.CAL oder .CNF) auswählen!"

- ➔ Der Pfad der Kalibrierdatei ist nicht korrekt.

"Kalibrierung wurde abgebrochen!"

"Daten sind nicht vollständig - Spektren nicht verwenden!"

- ➔ Abbruch oder Beenden Button wurde betätigt

"Kein Spektrum ausgewählt!"

- ➔ Zu kalibrierende Spektren müssen ausgewählt werden

## 6.1.4. Standardanalyse

Mit der Subroutine Standardanalyse kann eine vordefinierte Analysesequenz auf alle ausgewählten Spektren angewendet werden.

Eingabedateien sind zu analysierende Spektren (.CNF) und Ausgabedateien sind die dazugehörigen Reportdateien (.RPT) mit gleichem Namen. Reportdateien werden am Ort der ausgewählten Spektren in einen Unterordner „reports“ abgelegt.

### 6.1.4.1. Benutzereingaben - Standardanalyse

Mit dem Button „Spektren auswählen“ sind alle Spektren auszuwählen, auf welche die Analysesequenz angewendet werden soll.

In „Analysesequenz“ muss die gewünschte Analysesequenz ausgewählt werden. Als Zielordner des Browser-Dialogs ist C:\GENIE2k\CTLFILLES\ definiert. In diesem Ordner muss bei Starten der Standardanalyse die gewünschte Analysesequenz (.ASF) abgelegt sein.

Vor dem Start der Analyse muss vom Benutzer sichergestellt werden, dass keine bereits erstellten Reportdateien im Ordner „reports“ vorhanden sind, die den gleichen Namen aufweisen wie Reportdateien, welche im Zuge der bevorstehenden Analyse erstellt würden.

### 6.1.4.2. Individuelle Analysesequenz

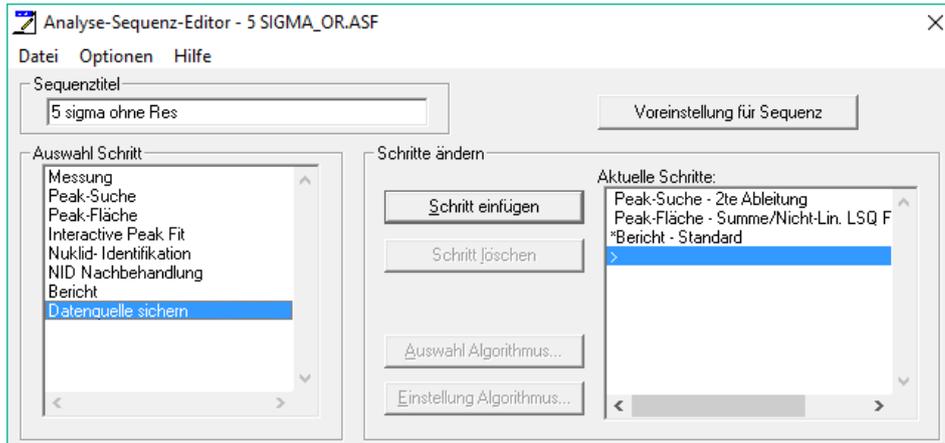
Um eine individuelle Analysesequenz zu erstellen, ist die schnellste Vorgehensweise der Aufruf des Analysesequenzen-Editors - Abbildung 31 - in der Kommandozeile durch den Befehl „ASE“.

Durch den Button „Schritt einfügen“ können verschiedenste Methoden aneinandergereiht werden. Mit den Buttons „Auswahl Algorithmus“ und „Einstellung Algorithmus“ können spezifische Parametereinstellungen getroffen werden.

Dies ermöglicht zum Beispiel eine Abfolge von Peaksuche- und Peakflächenmethoden mit angepassten Analyseparametern, ähnlich der Subroutine „Spezialanalyse“, welche mit Klassenfunktionen realisiert wurde.

Wird eine andere Vorlagendatei zur Erstellung von Reportdateien benötigt, kann eine neue Analysesequenz erstellt werden, in welcher das gewünschte Template im Schritt „Bericht - Standard“ mit „Einstellung Algorithmus“ ausgewählt werden kann.

Weitere Beispiele und Erklärungen zur Verwendung des Analyse-Sequenz-Editors sind im Genie-Manual [26] und [31] ausgeführt.



**Abb.31:** Analysesequenz „5 SIGMA\_OR“ dargestellt im Analysesequenz-Editor; Screenshot des Verfassers

### 6.1.4.3. Durchführung der Standardanalyse

Vor Start der Analyse muss Genie2000 geschlossen werden.

Wurden entsprechende Analysesequenzen erstellt und zu analysierende Spektren ausgewählt, kann die Standardanalyse durchgeführt werden. Abhängig von der Anzahl der Spektren und der Komplexität der Analysesequenzen kann dieser Vorgang einige Minuten dauern. Während der Analyse der Spektren wird der Fortschrittsbalken, ersichtlich in Abbildung 29 unter dem Button „Spektren auswählen“ regelmäßig aktualisiert, wobei der Rest der GUI, mit Ausnahme der Buttons „Abbrechen“ und „Beenden“ deaktiviert wird.

### 6.1.4.4. Fehlerquellen und -meldungen der Standardanalyse

- Die Spektren wurden mit einer Methode des Programms bereits analysiert und die Reportdateien der vergangenen Analyse befinden sich in den Zielordnern der bevorstehenden Messung. Da die Reportdateien nicht überschrieben werden sollen, wird die Analyse nicht gestartet.
- Während der Analyse darf keine weitere Instanz des Programms DUAL\_LFC gestartet werden, um mögliche Komplikationen zu vermeiden.
- Während der Analyse darf keines der zu analysierenden Spektren in Genie2000 geöffnet sein, da ansonsten der Zugriff und die Analyse des betreffenden Spektrums verweigert werden.

„Reportfile ist im Zielordner bereits vorhanden! Analysenabbruch.“

„[Zielreportpfad]“

- ➔ Im Ordner [Zielreportpfad] befindet sich eine Reportdatei mit gleichem Namen, wie die bevorstehende Analyse erstellen würde.

"Analysesequenz (.asf) existiert nicht!"

- ➔ Analysesequenz ist im Ordner C:\GENIE2k\CTLFILDES\ nicht vorhanden

"Analyse wurde abgebrochen!  
"Daten sind nicht vollständig - Reportfiles nicht verwenden!"  
→ Abbruch oder Beenden Button wurde betätigt

"Kein Spektrum ausgewählt!"  
→ Zu analysierende Spektren müssen ausgewählt werden

## 6.1.5. Spezialanalyse

Die Spezialanalyse dient zur Analyse von Spektren mit angepassten Analyseparametern. Die Parameter werden ausführlich in Kapitel 4.4.2.3. beschrieben.

Eingabedateien sind zu analysierende Spektren (.CNF) und Ausgabedateien sind die dazugehörigen Reportdateien (.RPT) mit gleichem Namen. Reportdateien werden am Ort der ausgewählten Spektren in einen Unterordner „reports“ abgelegt.

### 6.1.5.1. Benutzereingaben - Spezialanalyse

Mit dem Button „Spektren auswählen“ sind alle Spektren auszuwählen, auf welche die Spezialanalyse angewendet werden soll.

Für die Spezialanalyse entscheidet die Wahl des Spektrentyps „mittel“ oder „lang“ den Umfang der Analyse. Der Spektrentyp richtet sich nach der Abklingzeit der Proben nach der Aktivierung - also ob die Messungen 5 oder 28 Tage nach der Aktivierung durchgeführt wurden.

Weiters ist die Auswahl einer Analyseparameterdatei notwendig, welche durch einen Browser-Dialog oder durch manuelle Eingabe durchgeführt werden kann.

### 6.1.5.2. Anpassung einer Analyseparameter-Datei

In C:\GENIE2k\CTLFILLES\ ist die Standarddatei der Analyseparameter peaks.txt zu finden. Diese Datei kann beispielsweise in den Stammordner der Probenreihe kopiert werden, um die Analyseparameter, der aktuellen Probenreihe spezifisch, abzuändern. Die Parameterdatei sollte nicht in die Ordner „mittel“ oder „lang“ kopiert werden, da am Ende der Analyse die tatsächlich verwendete Parameterdatei an diesem Ort, zu Dokumentationszwecken, abgelegt wird. Eine bestehende Datei wird überschrieben. Vor Start der Analyse muss die entsprechende Datei ausgewählt sein.

Dazu muss bedacht werden, dass das Format der folgenden Parametertabelle, nicht verändert werden darf.

```
energie,kanal_anf,kanal_end,schwelle,signifikanz(2.abl),FIT
mittel:
103.2,241,280,5,1,0
208.4,532,551,2,1,0
277.4,713,747,2,1,0
559.1,1427,1466,1.5,2,0
684.7,1773,1794,2,2,1
1369,3536,3561,5,5,1
```

**Tab.14:** Ausschnitt der Parametertabelle peaks.txt, dargestellt im Editor

Soll eine weitere Energielinie während der mittleren oder langen Spezialanalyse analysiert werden, müssen die Parameter Energie, Kanalanzahl, Kanalende, Schwelle der Residuumsuche für die Peakflächenintegration, Schwelle der Signifikanz der Peaksuche mit 2. Ableitung und Fitte Einzellinien, durch Komma getrennt, als zusätzliche Zeile im entsprechenden Abschnitt „mittel:“ oder „lang:“, eingetragen werden. Als Dezimalzeichen ist ein Punkt zu verwenden und es dürfen keine Leerzeilen eingefügt werden.

Es ist vorteilhaft, dass neue Energielinien richtig, sortiert nach der Energie, in die Parametertabelle eingegliedert werden. Die Reihenfolge der Zeilen der Parametertabelle definiert die Reihenfolge der Analyse und dementsprechend die ausgegebenen Reportdateien. Ist die Parametertabelle nicht nach Energie sortiert, sind die analysierten Peaks in der Reportdatei ebenfalls unsortiert und unübersichtlich.

Sollen zusätzlich analysierte Energien auch ausgewertet werden, müssen weitere Modifikationen - Kapitel 6.1.7.3 - verrichtet werden.

Das Hinzufügen eines weiteren Parameters wird in Kapitel 6.2.2.3. beschrieben.

### **6.1.5.3. Durchführung der Spezialanalyse**

Vor Start der Analyse muss Genie2000 sowie die Datei der Analyseparameter geschlossen werden.

Sind Spektren und eine passende Parametertabelle ausgewählt sowie die Vorwahl getroffen, kann die Spezialanalyse durchgeführt werden. Wird die Sequenz nicht mit dem AUTOMatik-Modus durchgeführt, wird das Analyseende in der Endanzeige unterhalb des Fortschrittbalkens angezeigt. Der Fortschrittbalken wird mit abgeschlossenen analysierten Spektren aktualisiert. Während der Analyse werden alle Teile der GUI, außer der Abbruch und der Beenden Button, deaktiviert.

### **6.1.5.4. Fehlerquellen und -meldungen der Spezialanalyse**

- Die Spektren wurden mit einer Methode des Programms bereits analysiert und die Reportdateien der vergangenen Analyse befinden sich in den Zielordnern der bevorstehenden Messung. Da die Reportdateien nicht überschrieben werden sollen, wird die Analyse nicht gestartet.
- Während der Analyse darf keine weitere Instanz des Programms DUAL\_LFC gestartet werden um mögliche Komplikationen zu vermeiden.
- Während der Analyse darf keines der zu analysierenden Spektren in Genie2000 geöffnet sein, da ansonsten der Zugriff und die Analyse des betreffenden Spektrums verweigert werden.
- Bei der Erstellung von Reportdateien treten gelegentlich Fehler auf. Diese äußern sich durch fehlende Zeilenumbrüche oder Parametercodes. Treten derartige Diskrepanzen auf, kann dies einerseits zur Formatfehlermeldung führen, oder das Pro-

gramm zum Absturz bringen. Ist dies der Fall muss die fehlerhafte Reportdatei gefunden und korrigiert werden.

- Zu kleine Werte der Parameter „Schwelle der Residuum-Suche“ und „Signifikanzschwelle“ können zur Fehlermeldung „One or more peaks were dropped due to Multiplet-deconvolution.“ führen. Eine Anpassung der Analyseparameter ist unumgänglich.
- Falsche Formatierung der Parametertabelle kann zum Abbruch des Programms führen. Die Parametertabelle darf beim Start der Analyse nicht geöffnet sein um Zugriffsprobleme zu vermeiden.

„Pfad der Parameterdatei existiert nicht!“

→ Pfad der Parametertabelle muss überprüft werden

"Analysenabbruch - Parametertabelle ist falsch formatiert:"

"Das Dezimalzeichen muss ein Punkt(.), Spaltentrennzeichen ein Komma(,) sein."

" Die Tabelle muss 6 Spalten aufweisen!"

→ Format der Parametertabelle muss angepasst werden. Leerzeilen dürfen nicht vorhanden sein.

"Reportfile ist im Zielordner bereits vorhanden! Analysenabbruch."

„[Reportdateipfad]“

→ Bestehende Reportdatei [Reportdateipfad] muss umbenannt oder entfernt werden.

"Spezielle Analyse wurde abgebrochen!"

"Daten sind nicht vollständig - Reportfiles nicht verwenden!"

→ Abbruch oder Beenden Button wurde betätigt

"Kein Spektrum ausgewählt!"

→ Zu analysierende Spektren müssen ausgewählt werden

## 6.1.6. Interactive-Peak-Fit

Der Interactive-Peak-Fit (IPF) dient zur Untersuchung von Spektren und manuellen Analysen einzelner Peaks.

### 6.1.6.1. Durchführung des IPF

Mit dem Button „Spektren auswählen“ sind alle Spektren auszuwählen, für welche der IPF aufgerufen werden soll. Durch die Eingabe einer Energie wird bei Aufruf des IPF der entsprechende Energiebereich angezeigt. Für diese Eingabe sind nur Zahlen erlaubt.

Während des IPF befindet sich dieser im Fokus und die GUI des Programms DUAL\_LFC ist deaktiviert. Mit dem IPF können beispielsweise die Grenzen eines problematischen Peaks justiert, der Peak gefittet und die zugehörige Peakfläche integriert werden. Genauere Funktionen des Interactive-Peak-Fit sind im Genie-Manuel [27] nachzulesen.

Wird der IPF beendet, wird eine Eingabeform aufgerufen, welche es dem Benutzer ermöglicht, durchgeführte Änderungen am Spektrum oder durchzuführende Korrekturen in der Reportdatei mittels einer Referenz zu dokumentieren.

Da die Referenz direkt in der CAM-Datei gespeichert wird, und die zur Verfügung stehenden Zeichen auf 128 begrenzt sind, sollten die Eingaben möglichst kurz formuliert werden.

### 6.1.6.2. Fehlerquellen und -meldung des IPF

- Das Spektrum darf bei Aufruf des IPF in Genie2000 nicht geöffnet sein, da der IPF ansonsten keinen Zugriff auf das Spektrum erhält und das Programm zum Absturz bringt.
- Sind mehrere Spektren ausgewählt, wird der IPF für jedes Spektrum aufgerufen.

"Kein Spektrum ausgewählt!"

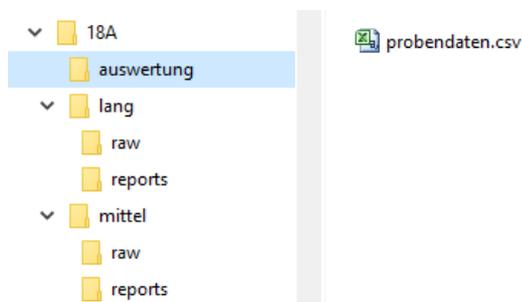
➔ Zu analysierende Spektren müssen ausgewählt werden

### 6.1.7. Auswertung

Die Auswertung vereint mittlere und lange Messungen einer Probenreihe in einer Komplettauswertung.

Eingabedateien sind die, durch die Analyse erstellten Reportdateien sowie *nuklide\_mittel.csv*, *nuklide\_lang.csv* und *probandaten.csv*, welche in Kapitel 6.4 näher beschrieben sind. Die Reportdateien müssen sich in den entsprechenden Ordnern „reports“ und die Datei *probandaten.csv* muss sich im Ordner „auswertung“, gemäß der Ordnerstruktur - beispielhaft mit Probenreihe 18A - in Abbildung 32, befinden.

Ausgabedateien sind *PRN\_logfile.csv*, *PRN\_messwerte.csv*, *PRN\_standards\_init.csv*, *PRN\_standards\_fin.csv* und *PRN\_auswertung.csv*, welche im Ordner „auswertung“ in Abbildung 32 abgelegt werden. PRN steht für Probenreihenname und spezifiziert die Dateinamen.



**Abb.32:** Ordnerstruktur vor Beginn der Auswertung;  
Screenshot des Verfassers

### 6.1.7.1. Benutzereingaben - Auswertung

Mit einem Browser-Dialog oder durch manuelle Eingabe muss der auszuwertende Stammordner der Probenreihe ausgewählt werden. In Abbildung 32 ist der Stammordner der Probenreihe „18A“.

Mit der Checkbox „Logfile erstellen“ wird definiert, ob aus allen analysierten CAM-Dateien die Referenzen ausgelesen und in einer Datei *PRN\_logfile.csv* ausgegeben werden. Diese Datei dient zu Dokumentationszwecken und als Übersicht der Referenzen. Dadurch kann sofort erkannt werden, wenn an einzelnen Spektren individuelle Maßnahmen gesetzt wurden.

Mit der Checkbox automatische Bearbeitung von Standards wird entschieden, ob die R-Werte der Standards aller Nuklide manuell vom Benutzer überprüft werden sollen oder ob die Entfernung von ungeeigneten R-Werten automatisch erfolgen soll.

### 6.1.7.2. Erstellung der Probendatei

Die Datei probendaten.csv muss im Ordner „auswertung“ abgelegt sein. Die enthaltene Tabelle muss drei Spalten, durch Komma getrennt, aufweisen.

```
name,masse[mg],standard
CFA,118.09,CFA
Pr-01,150.05,0
```

Tab.15: Zwei Proben aus probendaten.csv, dargestellt im Editor

Die erste Spalte beinhaltet die Bezeichnung der Probe, die zweite Spalte die eingewogene Masse in Milligramm und die dritte Spalte definiert, ob die Probe als Standard ausgewertet werden soll. Soll die Probe als Standard ausgewertet werden, muss in der dritten Spalte die korrekte Bezeichnung des Standards, wie in den Nukliddateien angegeben, definiert sein. Soll die Probe nicht als Standard ausgewertet werden, muss die dritte Spalte eine Null „0“ enthalten. Es dürfen keine Leerzeilen eingefügt werden und die Überschriften der Tabelle werden nicht eingelesen.

### 6.1.7.3. Anpassung der Nukliddateien

Sollen weitere Nuklide in die Auswertung miteinfließen, müssen die entsprechenden Energielinien bei der Analyse inkludiert worden sein (Kapitel 6.1.5.2.). Wurde die Parametertabelle erweitert, können gewünschte Nuklide in die Nukliddateien eingegliedert werden.

Zur besseren Übersichtlichkeit sollen neue Nuklide nach Energie sortiert in die Nuklidtabelle eingetragen werden.

Es ist vorteilhaft die Eintragung in Excel vorzunehmen und anschließend das Format korrekt anzupassen, um Fehler zu vermeiden. Überschriften, speziell die Bezeichnungen der Standards sind für den Ablauf der Auswertung essentiell, da die Konzentrationen der Standards durch die Bezeichnungen der Standards verglichen werden. Gleichmaßen

können auch neue Standards angefügt werden. Dabei ist darauf zu achten, dass die Bezeichnungen in nuklide\_mittel.csv und nuklide\_lang.csv ident sind.

```
Nuklid,HWZ (s),Energie (keV),E_interval,SO-1,CFA,LSS,GBW,SAT5,Ferrosilokon,IMS,Bonn  
Ce-141,2808090,145.4,1,102,190,72.07,163,58.91,,123.4,86.72  
Yb-169,2767046,177.2,1,2.24,7.6,3.12,4.51,5.032,,3.2  
Lu-177,581818,208.4,1,0.31,1.2,0.45,0.67,0.74,,0.468  
Pa-233 (Th),2329950,312.2,1,12.4,25.7,11.11,27.1,19.19,,20.51,15.64
```

Tab.16: Ausschnitt aus nuklide\_lang.csv, dargestellt im Editor

#### 6.1.7.4. Durchführung der Auswertung

Damit die Auswertung problemlos bewältigt werden kann, müssen die verwendeten Reportdateien einigen Anforderungen entsprechen.

Damit Reportdateien eingelesen werden können, muss der Abschnitt „Header“ 20 Zeilen betragen. Mit der 21. Zeile muss die Datentabelle beginnen, welche inklusive der Nummerierung der Peaks sieben Spalten, getrennt durch Leerzeichen, aufweisen muss. Die zweite Spalte muss die Energie, die dritte die Peakfläche und die siebte den Fehler der Peakfläche enthalten. Diesen Anforderungen genügen natürlich die Reportdateien der Spezialauswertung. Zusätzlich können Reportdateien, erstellt durch die Standardanalyse mit z.B. „5 SIGMA\_OR“, welche nach der Vorlagendatei a\_heinz.tpl erzeugt wurden, mit DUAL\_LFC ausgewertet werden. Reportdateien älterer Analysen können mit dieser Auswertung nicht ausgewertet werden, da in denen zwei Zeilen für die Referenz fehlen und der Header dementsprechend nur 18 Zeilen aufweist. Kompatibilitätsprobleme ergeben sich auch, wenn Reportdateien der neuen Analyseverfahren mit der bisherigen Auswertungsmethode ausgewertet werden sollen, da die VB-Routine, welche die Daten aus den Reportdateien in das Auswertung Excel „NAA\_Auswertung.xls“ überträgt ebenfalls auf die Größe des Headers als Fixpunkt zurückgreift. Diese stimmen aber nicht mehr überein.

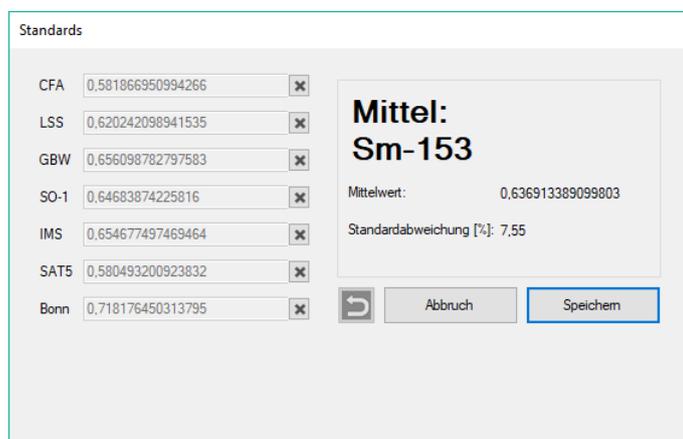
Es ist darauf zu achten, dass Verwechslungsgefahr besteht, da die Vorlagen a\_heinz\_livetime.tpl und a\_heinz\_realttime.tpl ebenfalls existieren. In diesen Vorlagen ist die Anzeige der Referenzen der Spektren nicht vorhanden und dementsprechend der Headerabschnitt zu kurz.

Wird das Logfile der Referenzen erstellt, kann die Auswertung, in Abhängigkeit von der Anzahl der Proben, einige Minuten in Anspruch nehmen, in welchen der Fortschrittsbalken aktualisiert wird.

Wird kein Logfile erstellt und wurde die **automatische Bearbeitung der Standards** gewählt, ist unter den oben beschriebenen Voraussetzungen die Auswertung nach einigen Sekunden abgeschlossen und es befinden sich im Ordner „auswertung“ alle Ausgabedateien.

Wird die **Bearbeitung der Standards manuell** durchgeführt, werden nacheinander alle Nuklide in der Standardform, abgebildet in 33, aufgerufen. Dem Benutzer obliegt nach Vergleich der R-Werte mit dem Mittelwert unter Berücksichtigung der Standardabweichung die Entscheidung ob und welche R-Werte entfernt werden sollen. Dabei müssen mindestens zwei R-Werte verbleiben, damit der Mittelwert berechnet werden kann. Ziel ist es die Standardabweichung akzeptabel anzupassen.

Dazu stehen die Entfernen-Buttons „x“ und der Rückgängig-Button zur Verfügung. Sind die R-Werte zur Zufriedenheit des Anwenders bearbeitet, müssen diese gespeichert werden.



**Abb.33:** Zwei Proben aus probendaten.csv, dargestellt im Editor; Screenshot des Verfassers

Umfasst eine Probenreihe mehr als zehn Standards, muss die Bearbeitung der Standards automatisch erfolgen, da die Standard-Form für maximal zehn Standards ausgelegt ist.

Ist die Bearbeitung abgeschlossen, werden folgende Ausgabedateien im Ordner "auswertung" erstellt, wobei die Dateinamen mit dem Namen der Probenreihe (PRN) spezifiziert werden:

- PRN\_logfile.csv:** Übersicht aller durchgeführten Analyseschritte
  
- PRN\_messwerte.csv:** Enthält alle Peakflächen und Fehler nach Einlesen der Reportdateien, zur Möglichkeit der Überprüfung der Analysequalität
  
- PRN\_standards\_init.csv:** Enthält Tabellen der Standard R-Werte, Mittelwerte, Standardabweichungen, Fehler der Mittelwerte - absolut, Fehler der Mittelwerte - relativ und Fehlerquadrate der Mittelwerte vor manueller oder automatischer Bearbeitung.
  
- PRN\_standards\_fin.csv:** Enthält Tabellen der Standard R-Werte, Mittelwerte, Standardabweichungen, Fehler der Mittelwerte - absolut, Fehler der Mittelwerte - relativ und Fehlerquadrate der Mittelwerte nach manueller oder automatischer Bearbeitung sowie eine Auflistung der entfernten R-Werte.
  
- PRN\_auswertung.csv:** Enthält die Endergebnistabelle, welche die Konzentrationen der Nuklide in den Proben und dazugehörige absolute Fehler enthält. Die Ergebnisse der mittleren und langen Messungen werden nebeneinander angeführt.

### 6.1.7.5. Fehlerquellen und -meldungen der Auswertung

- Ausgabedateien sind durch eine bereits durchgeführte Auswertung bereits vorhanden.
- Eingabedateien sind nicht vorhanden.
- Falsche Bezeichnungen der Standards in probendaten.csv führen zu Fehlern, wodurch das Programm DUAL\_LFC unterbrochen wird. Die Namen müssen korrigiert werden.
- Ungeeignete oder falsch formatierte Reportdateien können zum Abbruch der Auswertung führen. Spezielle Analysefehler können zu außergewöhnlichen Fehlermeldungen führen. Tritt eine nicht behandelte Fehlermeldung auf, muss diese in der Reportdatei manuell korrigiert werden, oder bei häufigerem Auftreten die automatische Behandlung in das Programm implementiert werden (Kapitel 6.2.2.).
- Änderungen der Vorlagedatei können zu Formatfehler der Reportdateien führen.

„Der Index war außerhalb des Arraybereichs.“, abgebildet in 34.

- ➔ Falsche Formatierung von Reportdateien, Probendateien oder Nukliddateien, insbesondere Leerzeilen, falsche Trenn- oder Dezimalzeichen

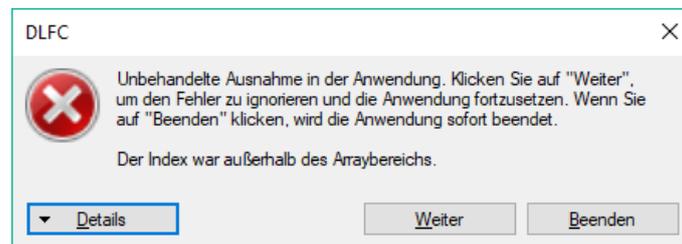


Abb.34: Fehlermeldung durch Formatfehler von Eingabedateien; Screenshot des Verfassers

"Auszuwertender Ordner existiert nicht!"

"Abbruch der Auswertung."

- ➔ Pfad des ausgewählten Stammordners überprüfen

"Reportfiles der mittleren/langen Messreihe nicht verfügbar!"

"Abbruch der Auswertung."

- ➔ Ordner „mittel“ und „lang“ sowie deren Unterordner „reports“ und benötigte Reportdateien überprüfen

"Fehlende Probendaten/Nuklidliste/! Abbruch der Auswertung."

"[Dateipfad]"

- ➔ Probendatei oder Nukliddatei am [Dateipfad] ist nicht vorhanden.

"Ausgabedatei ist im Zielordner bereits vorhanden! Abbruch der Auswertung."

"[Dateipfad]"

- ➔ Ausgabedatei am angegebenen [Dateipfad] ist bereits vorhanden und soll nicht überschrieben werden.

"Unvollständige Reportfiles."

"Anzahl der Reportfiles stimmen nicht mit Probandaten überein oder ist ungerade."

"Abbruch der Auswertung!"

- ➔ Anzahl der Reportdateien ist nicht doppelt so groß wie die Probenanzahl in Probandatei. Beides muss überprüft und angepasst werden.

"Abbruch der Auswertung - Formatfehler in Reportfile:"

"Bsp.: M/m/F werden abgeschnitten."

"Restliche Zeile muss 7 Spalten, mit Leerzeichen getrennt, enthalten!"

- ➔ Reportdateien müssen auf Unregelmäßigkeiten überprüft werden.

"Abbruch der Auswertung. Daten nicht vollständig."

- ➔ Abbruch oder Beenden Button der DLFC\_index-Form oder einer Standard-Form wurde betätigt.

„Namen der Standards in probendaten.csv stimmen nicht mit denen in"

"nuklide\_lang.csv überein!"

- ➔ Standardbezeichnungen müssen in probendaten.csv, nuklide\_mittel.csv und nuklide\_lang.csv angeglichen werden.

### **6.1.8. Automatische Mess- /Analyse- und Auswertungssequenz**

Die automatische Durchführung der Messung, Analyse und Auswertung ist nur für die lange Messung sinnvoll, da für die Auswertung die Reportdateien der mittleren Messung ebenfalls vorhanden sein müssen. Alle Vorbereitungen der Kapitel Messung, Analyse und Auswertung müssen getroffen sein.

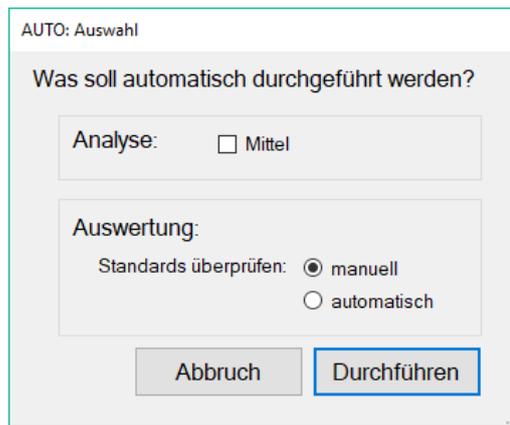
Eingabedateien sind die Spektren oder Reportdateien der mittleren Messung. Ausgabedateien sind Spektren und Reportdateien der langen Messung sowie alle Ergebnisse der Auswertung.

#### **6.1.8.1. Benutzereingaben - AUTO-Sequenz**

Als Messvorwahl muss „lang“ gewählt sein. Ansonsten müssen wie in Kapitel Messung die Parameter Probenanzahl, Messdauer und Speicherort gewählt werden. Der Name der Probenreihe muss mit dem bereits existierenden Stammordner am Speicherort übereinstimmen. Darin müssen vor dem Start der Sequenz die Unterordner „mittel“ und „auswertung“ vorhanden sein, wobei die gemessenen mittleren Spektren ebenfalls enthalten sein müssen.

Im Ordner „auswertung“ muss die Probandatei korrekt erstellt vorhanden sein. Die zu erstellenden Teile der Ordnerstruktur sind in Abbildung 32 ersichtlich.

Sind alle Benutzereingaben und Vorbereitung abgeschlossen, kann der Button „AUTO“ betätigt werden, wodurch die, in Abbildung 35 dargestellte, Methodenauswahl geöffnet wird.



**Abb.35:** GUI - Funktionsauswahl der automatischen Durchführung;  
Screenshot des Verfassers

Sind die Spektren der mittleren Messung noch nicht analysiert worden, muss die dazugehörige Checkbox ausgewählt werden. Zusätzlich muss definiert werden, ob die Bearbeitung der Standards, beschrieben in Kapitel 6.1.7.4., automatisch oder manuell erfolgen soll.

### 6.1.8.2. Durchführung der AUTO-Sequenz

Vor Start der AUTO-Sequenz muss Genie2000 sowie alle Reportdateien und Eingabedateien geschlossen werden.

Wurden die Parameter ausgewählt, kann die AUTO-Sequenz durchgeführt werden und die lange Messung der Proben startet.

Wie gehabt wird das Messende in „Aktuelle Messung“ angezeigt. Zudem wird das Label „AUTO-Mode: ON“ gesetzt und das voraussichtliche Ende der gesamten Sequenz im Label „AUTO-Mode - Ende:“ angezeigt.

Während der Analysesequenzen wird der Fortschrittsbalken aktualisiert.

Wurde automatische Bearbeitung der Standards während der Auswertung gewählt, wird die AUTO-Sequenz bis zum Ende durchgeführt. Wird eine manuelle Bearbeitung der Standards bevorzugt, wird die AUTO-Sequenz durchgeführt bis die Standard-Form des ersten Nuklids aufgerufen wird. Danach ist vom Anwender die Bearbeitung der R-Werte durchzuführen und die AUTO-Sequenz damit abzuschließen.

### 6.1.8.3. Fehlerquellen und -meldungen der AUTO-Sequenz

- Falsche oder fehlende Ordnerstruktur sowie fehlende Ressourcen der mittleren Messung.
- Nicht übereinstimmende Werte der Benutzereingaben zu Ressourcen der mittleren Messung oder Probendatei.

"Fehlende Parameter!"

➔ Benutzereingaben überprüfen

"Speicherort existiert nicht!"

- ➔ Pfad des ausgewählten Stammordners der Probenreihe und zugehörigen Probenname überprüfen

"Probendaten nicht vorhanden!"

"[Dateipfad]"

- ➔ [Dateipfad] der Probendatei überprüfen

"Es ist nur eine lange Messung sinnvoll."

"Spektren(.cnf) der mittleren Messung müssen im selben Ordner vorhanden sein."

"Probendaten.csv muss im Ordner [Probendatenpfad] vorhanden sein und Anzahl der Proben muss mit Anzahl der mittleren Spektren übereinstimmen!"

"Abbruch der automatischen Messung."

- ➔ Benutzereingaben und Daten der mittleren Messung müssen überprüft werden.

### 6.1.9. Referenzen anzeigen

Mittels des Button „ID-Key anzeigen“ werden für alle ausgewählten Spektren, die darin gespeicherten Referenzen, dargestellt in Abbildung 36, angezeigt.

Um Manipulationen zu vermeiden, welche die Nachvollziehbarkeit der Analyseschritte verringert, können Referenzen nicht bearbeitet oder gelöscht werden.

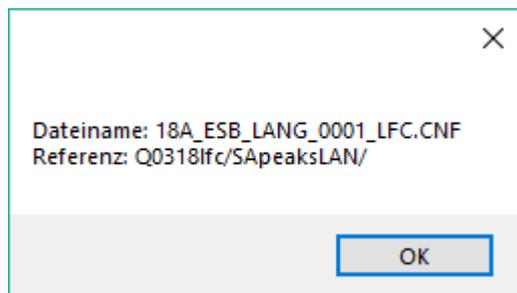


Abb.36: Darstellung der Funktion „ID-Key anzeigen“; Screenshot des Verfassers

## 6.2. IMPLEMENTIERUNG

### 6.2.1. DUAL\_LFC - Setup

Ist die Verwendung des Programms an einem weiteren Messsystem oder einem neuen Mess-PC vorgesehen, sind einige Vorbereitungen durchzuführen.

Für neue Messsysteme und Messrechner müssen alle Schritte durchgeführt werden, welche auch im Kapitel 3.2.3. abgearbeitet wurden. Insbesondere sollte darauf geachtet werden die gleiche Version des .NET-Frameworks zu verwenden, da die Verwendung des Programms DUAL\_LFC für diese Ziel-Framework-Version kompiliert wurde.

Bevor das Programm verwendet werden kann, müssen die nötigen Ressourcendateien, für die Methoden des Programms, erstellt und in den korrekten Quellordnern abgelegt sein.

Grundsätzlich muss ein Stammverzeichnis für das Programm erstellt werden. Damit dieses auf den Messsystemen ident ist, muss C:\Program Files\AutoMeasure\_DLFC\ erstellt und DUAL\_LFC.exe darin abgelegt werden. Das Erstellen einer Verknüpfung am Desktop zu dieser Datei ist vorteilhaft.

Für die **Messmethode** ist es erforderlich die ausführbare Datei DLFC\_ADV.exe, welche den Probenwechsler ansteuert, ebenfalls im Stammverzeichnis abzulegen. Diese Routine benötigt als Ressource die Datei inpout32.dll, welche im Stammverzeichnis abzulegen ist.

Es ist günstig einen Default Ordner - z.B. AUSWERTUNG als Speicherort für die Erstellung der Spektren während der Messung zu erstellen. Dieser sollte vorzugsweise auf einer Festplatte platziert sein, welche nicht das Betriebssystem enthält.

Die **Analysemethoden** benötigen die Ressourcendateien peaks.txt, a\_heinz.tpl und sanal.tpl, welche am Speicherort C:\GENIE2k\CTLFILLES\ abgelegt werden müssen.

Für die **Auswertungsmethode** müssen ebenfalls in C:\GENIE2k\CTLFILLES\ die Dateien nuklide\_mittel.csv und nuklide\_lang.csv zu finden sein.

All diese Ressourcen werden in einem Archiv gepackt in C:\Program Files\AutoMeasure\_DLFC\ und auf CD gesichert.

## 6.2.2. Modifizierung von Programmteilen

Um Erweiterungen der generellen Methoden am Quellcode durchführen zu können, sind vertiefende Programmierkenntnisse in VisualBasic oder C# notwendig. Darüber hinaus muss der Aufbau des Programms, beschrieben in Kapitel 4, grundlegend verstanden sein. Der Quellcode des Programms ist für das Verständnis des Aufbaus des Programmes überaus detailliert kommentiert.

Für detaillierte Modifizierungen der Mess- oder Analysemethoden sind Kenntnisse über die Funktion von COM-DLL Bibliotheken und Klassen unumgänglich. Um Basismethoden einzugliedern stehen Prozeduren zur Verfügung, welche durch Aufruf mittels BATCH-Befehlen gestartet werden können.

### 6.2.2.1. Allgemeine Veränderungen

Werden die Funktionen der Klassen Messung, Analyse oder Auswertung erweitert, müssen die Laufzeiten der entsprechenden Programmteile gemessen werden, um die Anzeigen Messende, Analyseende oder AUTO-Mode-Ende anzupassen.

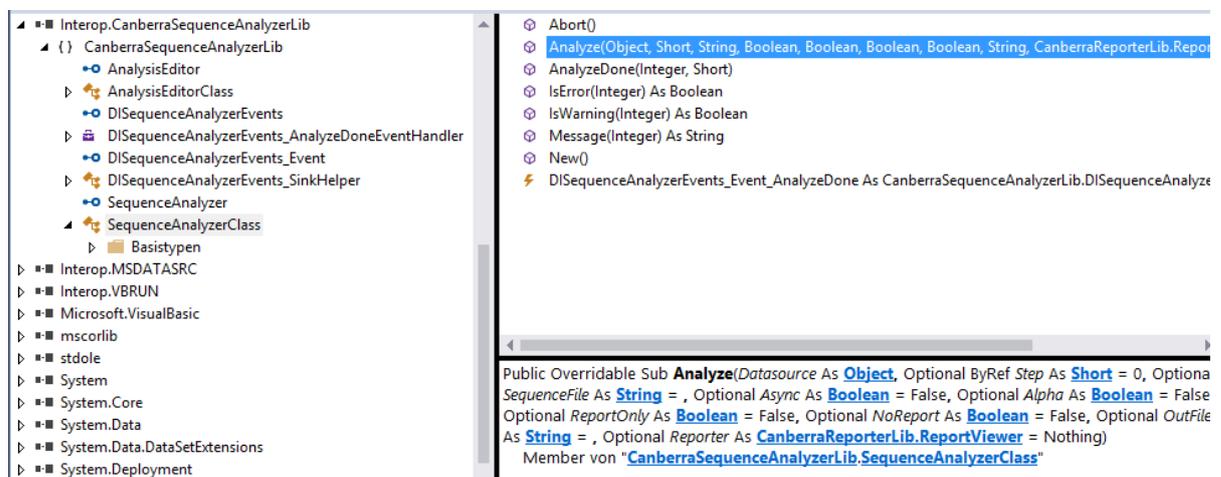
Gleiches gilt bei starken Veränderungen der Injektions- und Extraktionszeiten der Proben im Probenwechsler. Ist die Dauer des gesamten Probenwechsels länger als 30 Sekunden muss der Aufruf der Delay-Funktion sleep\_fct() angepasst werden.

Wenn weitere Eingabedateien eingelesen werden, ist unbedingt auf die Region-Einstellungen des Mess-PC's und die damit verbundene Formatierung der Dateien zu achten. Um Fehler zu vermeiden, sollten nur Textdateien eingelesen werden, welche ebenfalls als Zeilentrennzeichen Kommata und als Dezimalzeichen Punkte verwenden.

### 6.2.2.2. COM-DLL's

Verwendet wurden die drei Bibliotheken CanberraDataAccessLib, CanberraReporterLib und CanberraSequenceAnalyzerLib.

Beispielhaft ist in Abbildung 37 der Objektkatalog von Microsoft Visual Studio dargestellt. Dieser beinhaltet alle verknüpften Bibliotheken des Projekts.



**Abb.37:** Darstellung der Syntax der Analysefunktion im Objektkatalog von Visual Studio; Screenshot des Verfassers

In Abbildung 37 ist die Bibliothek CanberraSequenceAnalyzerLib und die Klasse SequenceAnalyzerClass dargestellt. Man erkennt welche Funktionen die Klasse enthält und am Beispiel von Analyze() kann die Syntax der Funktion betrachtet werden.

In Zuge der Installation von Canberra Genie2000 sind einige Bibliotheken inkludiert, welche in C:\GENIE2k\EXEFILES\ zu finden sind.

Sollen Modifizierungen durchgeführt werden, welche die Eingliederung weiterer Bibliotheken erfordern, muss im Projekt ein Verweis für jede benötigte Bibliothek hinzugefügt werden.

### 6.2.2.3. BATCH-Befehle

Einige Basisroutinen stehen als ausführbare EXE-Dateien zur Verfügung. Viele davon können bei Aufruf der Routinen mittels Attribute modifiziert werden. Alle BATCH Prozeduren sind detailliert im Genie-Manual [29] beschrieben.

Als Beispiel wird die Syntax des BATCH-Befehls PEAK\_DIF [3], welche das Analogon der „Peaksuche nach 2. Ableitung“ ist, angeführt.

PEAK\_DIF [Spektrenpfad] [/Qualifier(s)]

[Spektrenpfad]      Verpflichtende Angabe; Der Pfad des zu analysierenden Spektrums

[/Qualifier(s)]      Optionale Angabe; Zur Verfügung stehen einige Parameter zur spezifischeren Peaksuche

**Beispiel:**

PEAK\_DIF [Spektrenpfad] /CHANNELS= [Kanalstart],[Kanalende]  
/SIGNIF=[Signifikanzschwelle]

Soll beispielsweise ein weiterer Parameter zur spezifischeren Analyse als Benutzereingabe ermöglicht werden, ist dieser Parameter in der Parametertabelle als zusätzliche Spalte für alle Energien einzutragen und der betreffende BATCH-Befehl in der Klasse Analyse und Funktion analyse\_spez(), zu erweitern. Zusätzlich muss die Einlesefunktion und zugehörige Variablenmatrix der Parametertabelle erweitert werden.

## 6.3. KURZANLEITUNG

### 6.3.1. Messung

- Erste Probe manuell mittels Probenwechsler in den Detektor einbringen, mit Genie2000 messen und kalibrieren. Kalibrierung am Detektor speichern. Genie2000 schließen.
- DUAL\_LFC starten. Probenanzahl, Messdauer, Speicherort, Name der Probenreihe und optionale Abklingzeit-Vorwahl definieren und Messung starten.

### 6.3.2. Kalibrierung

- Spektren [Spektren auswählen], die zu übertragende Kalibrierdatei (.CAL/.CNF) sowie Kalibrierung auswählen.
- Kalibrierung durchführen.

### 6.3.3. Standardanalyse

- Spektren [Spektren auswählen] und gewünschte Analysesequenz (.ASF) auswählen.
- Standardanalyse durchführen.

### 6.3.4. Spezialanalyse

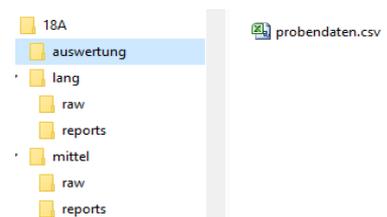
- Spektren [Spektren auswählen], Typ der Messung („mittel“ oder „lang“) sowie Parametertabelle (.TXT) auswählen.
- Spezialanalyse durchführen.

### 6.3.5. IPF

- Spektren [Spektren auswählen] und optional Energie definieren.
- IPF aufrufen. Nach erfolgtem IPF kann individuelle Referenz angefügt werden.

### 6.3.6. Auswertung

- Ordnerstruktur nach nebenstehendem Beispiel erstellen.
- Probendatei erstellen und in „auswertung“ ablegen
- Parameter „Logfile erstellen“ und „manuelle oder automatische Bearbeitung der Standards“ definieren
- Auswertung starten.
  - Manuelle Bearbeitung:  
R-Werte der Standards kontrollieren, Ungeeignete entfernen und Standards speichern.



**Abb.38:** Ordnerstruktur der Probenreihe 18A vor Beginn der Auswertung

**Ausführlichere Beschreibungen der Methoden des Programms DUAL\_LFC sind in der Bedienungsanleitung, am Ort C:\Program Files\AutoMeasure\_DLFC\, zu finden.**

## 7. CONCLUSIO

Die Motivation dieser Arbeit wurde anfangs anhand der folgenden drei Hauptkomponenten zusammengefasst:

1. Die Neuorganisation des Messaufbaus.
  2. Ein anwenderfreundliches Programm erstellen, welches die Schritte der Messung, Analyse und Auswertung der INAA vereint und die Qualität der Prozesse steigert.
  3. Die Sicherstellung der Reproduzierbarkeit und Erhöhung der Datensicherheit.
- 1 - Durch die Modernisierung der Messsysteme und Angleichung der Hard- und Software der beiden Messrechner wird die Bedienung der Messsysteme effizienter. Die identischen Spezifikationen des PC2 sowie das Programm DUAL\_LFC ermöglichen nun die Modifikation des „Russen“ mit einem Probenwechslersystem. Demnach können gleichzeitig zwei Probenreihen gemessen werden, was zu einem höheren Durchsatz der INAA führt.
- 2 - Die Verbesserungen und Verschlechterungen der einzelnen Prozessschritte der INAA sind ausführlich in Kapitel 5 ausgeführt. Wesentlichen Veränderungen sind folgend angeführt:
- Die einheitliche Benutzeroberfläche für die Teile der INAA ermöglicht eine intuitive Anwendung des Programms. Eine umfassende Fehlerbehandlung von möglichen Programmabläufen erleichtert die Bedienung.
  - Durch die Anpassung der Analyseparameter ist eine markante Verbesserung der Registrierung von Energielinien im Spektrum ersichtlich. Energielinien bei etlichen Proben, welche mit der bisherigen Analysemethode nicht gefunden wurden, werden nun registriert. Dieser Effekt ist bei allen Nukliden ersichtlich, tritt aber verstärkt bei Wolfram, Neodym und Zirkonium auf. Es muss bedacht werden, dass aufgrund des oft schlechten Intensitäts-zu-Hintergrund-Verhältnisses der Fehler derartiger Peaks meist groß ist.
  - Die Energielinien der Nuklide Samarium, Lutetium und Neptunium mussten mit bisheriger Auswertung in mühevoller Kleinarbeit für die Proben der gesamten Probenreihe manuell gefittet werden, um die Peakflächen einzeln korrigieren zu können. Mit der Methode der Spezialanalyse kann diese Arbeit stark reduziert werden. Quantitativ kann für die Probenreihe 17G festgestellt werden, dass für Samarium keine Nachbesserung nötig ist, für Lutetium bei zehn Proben korrigiert werden muss und bei vier Proben die Linie von Neptunium manuell nachgebessert werden muss. Dies ist eine Verbesserung von 91 % allein bei diesen drei Nukliden.
  - Anzumerken ist, dass den Nukliden Sm, Lu, Np weiterhin Aufmerksamkeit gewidmet werden sollte, um Änderungen der Qualität zu bemerken und sinnvoll einzusetzen, indem z.B. Parameter neu justiert werden.
  - Die Qualität der Analyse kann in folgenden Probenreihen für alle Nuklide noch weiter gesteigert werden. Dazu sollten sukzessive die Parametereinstellungen verfeinert

werden, bis möglichst effiziente Einstellungen für die Peaksuche niedrig intensiver Peaks, für die Residuum-Suche von Peaks in Multipletts und die Peakflächenintegration, gefunden wurden.

- Die Auswertung einer gesamten Probenreihe mit DUAL\_LFC erfolgt äußerst effektiv. Die Auswertung kann anhand Reportdateien der bisheriger Analyseverfahren „5 SIGMA\_OR“ oder anhand neuer Reportdateien, welche aus der „Spezialanalyse“ resultieren, durchgeführt werden. Dies eröffnet die Möglichkeit die Ergebnisse alter und neuer Analyse- und Auswertungsmethoden zu vergleichen und Unterschiede herauszuarbeiten.
- Die Auswertung definiert sich stark durch die Qualität der R-Werte der Standards und deren Mittelwert. Einerseits ist eine Qualitätssteigerung der R-Werte durch die verbesserte Analyseverfahren zu verzeichnen, andererseits ist durch die manuelle und automatische Bearbeitung der R-Werte eine Methode implementiert, um die Standardabweichung zu senken und die Qualität des Mittelwertes zu erhöhen. Wird die Bearbeitung manuell ausgeführt, ist dies zwar zeitaufwendiger, doch die Genauigkeit der Ergebnisse lässt sich signifikant steigern.

3 - Zur Sicherstellung der Reproduzierbarkeit wurde ein enormer Aufwand betrieben und erhebliche Änderungen eingeführt.

- Da während der gesamten INAA häufig optionale Arbeitsschritte und die Anwendung verschiedener Methoden möglich sind, musste ein Konzept entwickelt werden, mit welchem durchgeführte Maßnahmen dokumentiert und gesichert werden. Dazu werden mit Spektren durchgeführte Methoden direkt im jeweiligen Spektrum referenziert. Diese Referenzen werden in der zugehörigen Reportdatei ausgegeben und während der Auswertung in einer Logdatei gesammelt. Dadurch wird eine Übersicht erstellt, in welcher sofort ersichtlich ist, an welchen Probenspektren spezielle Maßnahmen durchgeführt wurden.
- Bei problematischen Nukliden sind bei manchen Proben weiterhin einige Peakflächen zu korrigieren. Dies stellt eine Verletzung der Reproduzierbarkeit dar und muss dementsprechend behandelt werden. Dazu wurde die Funktion des Interaktiven-Peak-Fit in das Programm implementiert. Mit diesem IPF können und sollen besagte Linien korrigiert werden. Dieser Schritt ermöglicht eine individuelle Dokumentation direkt in der Referenz der Spektren, wobei derartige Änderungen bei der Auswertung sofort ersichtlich sind.
- Es werden immer alle benötigten veränderlichen Parameter, welche für die Exekution einer Methode verwendet werden, dokumentiert und am Speicherort der Probenreihe gesichert. Weiters werden die wichtigsten Zwischenergebnisse der Auswertung separat in Ausgabedateien gesichert, um den Entstehungsprozess transparent zu gestalten. Diese Ausgabedateien werden ebenfalls im Auswertungsordner der Probenreihe abgelegt. Daraus entsteht für jede Probenreihe ein komplettes Paket aller variablen Eingabedateien, Ausgabedateien und Parameter.
- Die automatisierte Dokumentation der Entfernung von R-Werten in entsprechenden, übersichtlichen Ausgabedateien, während der manuellen oder automatischen Bear-

beitung der Standard-R-Werte, stellt die Nachvollziehbarkeit und Reproduzierbarkeit dieser Maßnahme sicher.

Zusätzlich ist zu erwähnen, dass durch den umfassend kommentierten Quellcode des Programms sowie der Kapitel 4 und 6 die Funktionalität von DUAL\_LFC transparent, vollständig und nachvollziehbar vermittelt wird, um die Erweiterung des Programms sowie Modifizierungen von Funktionen zu erleichtern.

## 8. LITERATUR

- [1] G. Steinhauser, J. H. Sterba, M. Bichler, H. Huber; "Neutron activation analysis of Mediterranean volcanic rocks – An analytical database for archaeological stratigraphy"; Applied Geochemistry 21; Atominstytut der technischen Universität Wien; 2006,
- [2] C. Stieghorst; "Neutronenaktivierungsanalyse in Archäometrie und Solarenergieforschung"; Dissertation; Kernchemie der Johannes Gutenberg-Universität Mainz; 2016
- [3] K. Lotter; "Anwendung der verdünnungsmodifizierten Mahalanobis-Distanz zur Auswertung von Keramikanalysen von der Insel Sai"; Bachelorarbeit; Atominstytut der technischen Universität Wien; 2015
- [4] D. Behne, D. Gawlik; "Spurensuche: Neutronen-Aktivierungsanalyse in der Spurenelementforschung"; Berichte aus dem HMI, 1/89; Hahn-Meitner-Institut, Helmholtz Zentrum in Berlin; 1989
- [5] L. Xiaosong; "Neutronenaktivierungsanalyse (NAA) an der RCM"; Radiochemie(RCM) München, TU München; „<https://www.rcm.tum.de/index.php?id=65&L=0>“, Letzter Zugriff: April 2018
- [6] A. Wieghaus, S. Ritzel; "Neutronenaktivierungsanalyse (NAA)"; Kernchemie der Philipps-Universität Marburg; „<https://www.staff.uni-marburg.de/~kernchem/grundp/naa/naa.htm>“, Letzter Zugriff: April 2018
- [7] W. Demtröder; "Experimentalphysik 4: Kern-, Teilchen-, Astrophysik"; 4.Auflage, 2013
- [8] K. R. Atkins; "Physik - Die Grundlagen des physikalischen Weltbildes"; University of Pennsylvania; 2. deutsche Auflage übersetzt von H.-W. Sichtung, 1986
- [9] K. Wartmann; "Überprüfung des Loss-free Gamma-Spektrometers auf Zählverluste bei mittleren bis hohen Zählraten"; Projektarbeit; Atominstytut der österreichischen Universitäten, TU Wien, 2006
- [10] G. F. Knoll; „Radiation Detection and Measurement“; University of Michigan, Ann Arbor; 3. Auflage 1999
- [11] K. Schmid; "Radiochemie und Gammaspektroskopie"; Projektarbeit; Atominstytut der technischen Universität Wien; 2001
- [12] B. Hindinger; "Zählverlustkorrektur eines Gammastrahlendetektors durch Kalibrierung des LFC-Systems"; Projektarbeit; Atominstytut der technischen Universität Wien; 2016
- [13] G. P. Westphal, F. Grass, H. Lemmel, J. Sterba, P. Schröder, C. Bloch; "Automatic activation analysis"; Journal of Radioanalytical and Nuclear Chemistry, Vol 271; Atominstytut der technischen Universität Wien; 2007
- [14] H. Mommsen; „Teilcheninduzierte Röntgenfluoreszenzanalyse und Neutronenaktivierungsanalyse in der Archäometrie“; Institut für Strahlen- und Kernphysik der Universität Bonn; 1987
- [15] T.Beier; „Korrektur- und Kontrollmechanismen bei der archäometrischen Neutronenaktivierungsanalyse“; Diplomarbeit; Institut für Strahlen- und Kernphysik der Universität Bonn; 1990

- [16] F. Chirigati<sup>1</sup>, M. Troyer<sup>2</sup>, D. Shasha<sup>3</sup>, J. Freire<sup>1,3</sup>; „A Computational Reproducibility Benchmark“; Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 2013; <sup>1</sup>Department of Computer Science and Engineering, New York University, <sup>2</sup>Institute for Theoretical Physics, ETH Zürich, <sup>3</sup>Courant Institute of Mathematical Sciences, New York University
- [17] J. Potthoff<sup>1</sup>, J. van Wezel<sup>1</sup>, M. Razum<sup>2</sup>, M. Walk<sup>1</sup>; „Anforderungen eines nachhaltigen, disziplinübergreifenden Forschungsdaten-Repositorys“; <sup>1</sup>Karlsruher Institut für Technologie (KIT), <sup>2</sup>FIZ Karlsruhe -Leibniz-Institut für Informationsinfrastruktur, 2014
- [18] J. S. Armstrong<sup>1</sup>, K. C. Green<sup>2</sup>; „Guidelines for Science: Evidence and Checklists“; ScholarlyCommons, “[http://repository.upenn.edu/marketing\\_papers/181](http://repository.upenn.edu/marketing_papers/181)”; <sup>1</sup>University of Pennsylvania, <sup>2</sup>University of South Australia; 2017, Letzter Zugriff: 29.4.2018
- [19] M. Baker, D. Penny; “Is there a reproducibility Crisis?”; Nature - Vol. 533, San Francisco; 2016
- [20] M. Hahn<sup>1</sup>, M. Razum<sup>1</sup>, J. Neumann<sup>2</sup>; „RADAR – Ein Forschungsdaten-Repository als Dienstleistung für die Wissenschaft“; Zeitschrift für Bibliothekswesen und Bibliographie, Februar 2014, <sup>1</sup>FIZ Karlsruhe -Leibniz-Institut für Informationsinfrastruktur, <sup>2</sup>Technische Informationsbibliothek (TIB) Hannover; 2014
- [21] S. Fomel<sup>1</sup>, J. F. Claerbout<sup>2</sup>; „Reproducible Research“; IEEE Jänner/Februar 2009, <sup>1</sup>The University of Texas at Austin, <sup>2</sup>Stanford University
- [22] R. D. Peng; „Reproducible Research in Computational Science“; Science, Dezember 2011, Department of Biostatistics, Baltimore
- [23] A. Casadevall<sup>1</sup>, F. C. Fang<sup>2</sup>; „Reproducible Science“; Infection and Immunity, American Society for Microbiology, Dec. 2010, <sup>1</sup>Albert Einstein College of Medicine New York, <sup>2</sup>University of Washington School of Medicine
- [24] Österreichische Agentur für wissenschaftliche Integrität; „Richtlinien der OeAWI zur guten wissenschaftlichen Praxis“; OeAWI, Sensengasse 1, 1090 Wien, 2010
- [25] G. K. Sandve<sup>1</sup>, A. Nekrutenko<sup>2</sup>, J. Taylor<sup>3</sup>, E. Hovig<sup>1</sup>; “Ten Simple Rules for Reproducible Computational Research”; Computational Biology, 2013, Vol 9, Issue 10; <sup>1</sup>University of Oslo, <sup>2</sup>Penn State University Pennsylvania, <sup>3</sup>Emory University Atlanta
- [26] Canberra Industries Inc; “Genie 2000 Operations Manual”, Meriden, USA; 2006
- [27] Canberra Industries Inc; “S506 Interactive Peak Fit User's Manual”, Meriden, USA; 2006
- [28] Canberra Industries Inc; “Genie 2000 Customization Tools Manual”, Meriden, USA; 2006
- [29] Canberra Industries Inc; “S561 Batch Tools Support”, Meriden, USA; 2006
- [30] Canberra Industries Inc; “Historical Canberra MCAs”, Meriden, USA; 2006
- [31] Canberra Industries Inc; “Genie 2000 Tutorials Manual”, Meriden, USA; 2006

## 9. ABBILDUNGS- UND TABELLENVERZEICHNIS

Abb.1:	Hahn-Meitner-Institut, Berlin, „ <a href="http://portal.uni-freiburg.de/janiak/Lehre/ac-pdf/vl22">http://portal.uni-freiburg.de/janiak/Lehre/ac-pdf/vl22</a> “
Abb.2:	[2]; Seite 9;
Abb.3:	[6]
Abb.4:	[9]; Seite 5
Abb.5:	Privatfoto; April 2018
Abb.6, 7:	Screenshot - Messrechner, April 2018
Abb.8:	Privatfoto; Dezember 2017
Abb.9-38:	Screenshot - Messrechner, April 2018
Tab.1:	Auswahl der Nuklide und Energien der mittleren Messung
Tab.2:	Auswahl der Nuklide und Energien der langen Messung
Tab.3:	AccuSpec B - Adresseinstellungen
Tab.4:	Parametertabelle - peaks.txt - für Spezialanalyse
Tab.5:	Auszug aus Eingabedatei - probendaten.csv
Tab.6:	Eingabedatei - nuklide_mittel.csv
Tab.7:	Auszug Messwertvergleich
Tab.8:	Vergleich von R-Werten bei Standard GBW
Tab.9:	Vergleich von R-Werten bei Standard SO-1
Tab.10:	Vergleich von Standardwerten - 17G, Mittel
Tab.11:	Vergleich von Standardwerten - 17G, Lange
Tab.12:	Standardkonzentrationsüberprüfung CFA - „ohne“
Tab.13:	Konzentrationsabweichungen - „ohne Standard“
Tab.14:	Ausschnitt der Parametertabelle peaks.txt
Tab.15:	Ausschnitt aus probendaten.csv
Tab.16:	Ausschnitt aus nuklide_lang.csv

### DANKSAGUNG

Für die Unterstützung bei der Erstellung und für die Möglichkeit diese Diplomarbeit am Atominstitut durchführen zu können möchte ich meiner Hauptbetreuerin Ao.Univ.Prof.Dipl.-Ing.Dr.techn. Christina Strelt besonders danken.

Meinem Nebenbetreuer Senior Scientist Dipl.-Ing.Dr.techn. Johannes Sterba möchte ich für die durchgehende Unterstützung und etlichen investierten Stunden, sowie für die gelegentlich erforderliche Motivation bei Problemstellungen sehr herzlich danken.

# ANHANG A - CANBERRA COM-DLL SYNTAX

Vollständige Syntax der Funktionen Open, Close, Save, Param, Spectrum und Analyze, extrahiert mit dem Objektkatalog von Visual Studio aus den COM-DLL's CanberraDataAccessLib, CanberraReporterLib und CanberraSequenceAnalyzerLib.

Public Overridable Sub **Open**(*bstrName* As [String](#), Optional *lOptions* As [CanberraDataAccessLib.OpenMode](#) = 0, Optional *lNumChans* As [Integer](#) = 0)  
Member von "[CanberraDataAccessLib.DataAccessClass](#)"

Public Overridable Sub **Close**(Optional *lOptions* As [CanberraDataAccessLib.CloseMode](#) = 4)  
Member von "[CanberraDataAccessLib.DataAccessClass](#)"

Public Overridable Sub **Save**(*Name* As [String](#), Optional *vbOverWrite* As [Boolean](#) = False)  
Member von "[CanberraDataAccessLib.DataAccessClass](#)"

Public Overridable Property **Param**(*lCodes* As [CanberraDataAccessLib.ParamCodes](#), Optional *lRec* As [Integer](#) = 1, Optional *lEntry* As [Integer](#) = 1) As [Object](#)  
Member von "[CanberraDataAccessLib.DataAccessClass](#)"

Public Overridable Property **Spectrum**(Optional *lStart* As [Integer](#) = 1, Optional *lStop* As [Integer](#) = -1, Optional *lRow* As [Integer](#) = 1, Optional *lGroup* As [Integer](#) = 1) As [Object](#)  
Member von "[CanberraDataAccessLib.DataAccessClass](#)"

Public Overridable Sub **Analyze**(*Datasource* As [Object](#), Optional ByRef *Step* As [Short](#) = 0, Optional *SequenceFile* As [String](#) = , Optional *Async* As [Boolean](#) = False, Optional *Alpha* As [Boolean](#) = False, Optional *ReportOnly* As [Boolean](#) = False, Optional *NoReport* As [Boolean](#) = False, Optional *OutFile* As [String](#) = , Optional *Reporter* As [CanberraReporterLib.ReportViewer](#) = Nothing)  
Member von "[CanberraSequenceAnalyzerLib.SequenceAnalyzerClass](#)"

Public Overridable Sub **ExecuteReport**(*Datasource* As [Object](#), *Template* As [String](#), Optional *Section* As [String](#) = <All>, Optional *Options* As [CanberraReporterLib.ReportOptions](#) = 9, Optional *Sigma* As [Single](#) = 1, Optional *ActivityUnits* As [String](#) = , Optional *ActivityMultiplier* As [Single](#) = 0)  
Member von "[CanberraReporterLib.ReportViewerClass](#)"

# ANHANG B - QUELLCODE DLFC\_ADV (PROBENWECHSLER)

```
program DLFC_ADV;
uses inpout,crt;                               {input: Input/Output; crt: Delay}

Var
  portnum:      integer;                        {Portnr.}
  portdata,portread: byte;                    {Datenpaket, ausgelesenes Datenpaket}
begin
  portnum:=53411;                               {Portnummer definieren}
  writeln('Port: ', portnum);                   {Ausgabe der Portnr.}
  portread:=inp32(portnum);                    {auslesen des Datenbytes am ang. Port durch aufrufen der Fkt inp32 aus input}
  writeln('ON/OFF: ', portread);               {Ausgabe des ausgelesenen Datenbyte}

  portdata:=255;                               {Setzen des Datenbyte auf 255 - Pin=1}
  writeln(' ');                                {Leerzeile}
  writeln('PIN gleich 1 setzen. ');             {AUSGABE: PIN aktivieren}
  out32(portnum,portdata);                    {Datenbyte am Port setzen (PIN aktivieren)}
  portread:=inp32(portnum);                   {Datenbyte am Port auslesen (TEST)}

  If portread > 0 then writeln('ON/OFF: 1');   {Testausgabe ob PIN gesetzt}
```

```

Delay(3000);                {beliebig lange Zeitverzögerung für PW in Millisekunden}

portdata:=0;                {Setzen des Datenbyte auf 0 - Pin=0}
writeln(' ');              {Leerzeile}
writeln('PIN gleich 0 setzen. '); {AUSGABE: PIN deaktivieren}
out32(portnum,portdata);    {Datenbyte am Port setzen (PIN deaktivieren)}
portread:=inp32(portnum);

If portread = 0 then writeln('ON/OFF: 0'); {Testausgabe ob PIN gesetzt}
end.

```

## ANHANG C - QUELLCODE DUAL\_LFC

Vollständiger Quellcode der Klassen DLFC\_index.vb, messung.vb, analyse.vb und auswertung.vb sowie deren Tochterklassen auto\_auswahl.vb, control\_box.vb, input\_box.vb und standards.vb.

### KLASSE - DLFC\_INDEX.VB

```

Public Class DLFC_index      {637 Zeilen}

'Variablen, welche alle Funktionen der Klasse verwenden
Dim i, j, k, l, m As Integer      'Schleifenindizes
Dim stat As String = ""          'Statusvariable
Public Shared error_num As Integer = 0      'Fehlervariable für ALLE Klassen
Dim camdata As New CanberraDataAccessLib.DataAccess      'neue Instanz der Canberra-DataAccess-Klasse

Dim anal_files() As String      'Spektronspeicherorte für Analyse und Kalibrierung
Dim amount As Integer          'Anzahl der Spektren
Dim laengenvorg As String      'Längenvorgabe zur Namensgebung der Spektren
Dim id_ref1 As String          'Reproduzierbare Durchführen - ID-Referenz 1
Dim id_ref2 As String          'Reproduzierbare Durchführen - ID-Referenz 2

Dim laufzeittest As DateTime

Private Sub auto_fct()
'Automatische Durchführung von Messung/Analyse/Auswertung

Application.DoEvents()
Dim auto_mode As Boolean = True      'Automatischer Modus aktiv
Dim mit_ok As Boolean = False        'Mittlere Messung initial NICHT in Ordnung
Dim filepath As String              'Dateipfad
Dim auto_date_var As Integer        'Datumvariable für Endanzeige der automatischen Durchführung

Dim frm_auto_auswahl As New auto_auswahl      'Form für Automatikmodus-Auswahl
Dim m_check As Boolean = False          'Mittlere Dateien Überprüfung
Dim std_ch As Boolean = False          'Automatische/Manuelle Bearbeitung der Standards

'---Überprüfe eingaben---
If (TBanzahl.Text = "" Or TBmessdauer.Text = "" Or
    TBpfad.Text = "" Or TBdateiname.Text = "") Then

    System.Windows.Forms.MessageBox.Show("Fehlende Parameter!")      '--> Fehler
    DLFC_index.error_num = 1
    GoTo LastLine_auto

Elseif (Not System.IO.Directory.Exists(TBpfad.Text)) Then
    System.Windows.Forms.MessageBox.Show("Speicherort existiert nicht!")      '--> Fehler
    DLFC_index.error_num = 1
    GoTo LastLine_auto

Elseif (Not System.IO.File.Exists(TBpfad.Text + TBdateiname.Text + "auswertung\probendaten.csv")) Then
    System.Windows.Forms.MessageBox.Show("Probendaten nicht vorhanden!" & vbCrLf &
        TBpfad.Text + TBdateiname.Text + "auswertung\probendaten.csv")
    DLFC_index.error_num = 1      '--> Fehler
    GoTo LastLine_auto
End If

'überprüfe Dateien der mittleren Messung
filepath = TBpfad.Text & TBdateiname.Text & "mittel"
If (System.IO.Directory.Exists(filepath)) Then      'Ordner-Mittel existiert
    If (Not IO.Directory.GetFiles(filepath, "*.cnf").Count = 0) Then
        If (IO.Directory.GetFiles(filepath, "*.cnf").Count = CInt(TBanzahl.Text) * 2) Then      'Dateien-Mittel existieren
            mit_ok = True      '--> mittlere Messung vorhanden
        End If
    End If
End Sub

```

```

End If
End If

'AUTO nur starten wenn mittlere Messung vorhanden und lange Messung ausgewählt
If (RBlang_acq.Checked And mit_ok) Then

'Auswahl-Form aufrufen
error_num = frm_auto_auswahl.auto_ausw_fct(m_check, std_ch)
If error_num = 1 Then GoTo LastLine_auto

'Teile der GUI für Benutzer deaktivieren
GBparameter.Enabled = False
GBcalibrate.Enabled = False
GBanalyse.Enabled = False
GBauswertung.Enabled = False
B_spektrenwahl.Enabled = False
Bid_key.Enabled = False
Bauto.Enabled = False
auto_date_var = (CInt(TBanzahl.Text) * (CInt(TBmessdauer.Text) + 30 + 5)) _
+ (CInt(TBanzahl.Text) * 240) 'ACHTUNG: anpassen wenn Delay verändert wird

L_stat_auto.Text = "AUTO-Mode: ON" 'Statusanzeige ändern
L_auto_endzeit.Text = DateAdd(DateInterval.Second, auto_date_var, Now) 'Ende der AUTO-Durchführung anzeigen
L_auto_ende.Visible = True 'Statusanzeige verbergen
L_auto_endzeit.Visible = True 'Statusanzeige verbergen

'-----lange MESSUNG-----
Call messen_fct(auto_mode)

'-----SPEZIALANALYSE-----
'---Mittlere Analyse---
If m_check = True Then
filepath = TBpfad.Text & TBdateiname.Text & "\mittel"
anal_files = IO.Directory.GetFiles(filepath, "*.cnf")
amount = anal_files.Count
RBmittel_ana.Checked = True
Call analyse_spez_fct(auto_mode)
End If

'---Lange Analyse---
filepath = TBpfad.Text & TBdateiname.Text & "\lang"
anal_files = IO.Directory.GetFiles(filepath, "*.cnf")
amount = anal_files.Count
RBlang_ana.Checked = True
Call analyse_spez_fct(auto_mode)

'-----AUSWERTUNG-----
filepath = TBpfad.Text & TBdateiname.Text & ""

If (std_ch = False) Then 'FALSE für MANUELLE Standardwahl
CBauto_std.Checked = False 'Wenn manuelle Standardauswahl gewählt --> Standards PopUP
Elseif (std_ch = True) Then 'TRUE für AUTOMATISCHE Standardwahl
CBauto_std.Checked = True 'Wenn automatische Standardauswahl gewählt --> kein Standard PopUP
End If

TB_ordner_ausw.Text = filepath
Call auswertung_fct(auto_mode)

Else
System.Windows.Forms.MessageBox.Show(
"Es ist nur eine lange Messung sinnvoll." & vbCrLf _
& "Spektren(.cnf) der mittleren Messung müssen im selben Ordner vorhanden sein." _
& vbCrLf & "Probandaten.csv muss im Ordner " & TBpfad.Text & TBdateiname.Text _
& "\auswertung\ vorhanden sein und Anzahl der Proben muss mit Anzahl der " _
& "mittleren Spektren übereinstimmen!" _
& vbCrLf & "Abbruch der automatischen Messung.")
error_num = 1
End If

LastLine_auto:
If (error_num = 0) Then System.Windows.Forms.MessageBox.Show _
("Automatische Messung/Analyse/Auswertung-Sequenz erfolgreich durchgeführt!")

'GUI aktivieren / Anzeigen verbergen
GBparameter.Enabled = True
GBcalibrate.Enabled = True
GBanalyse.Enabled = True
GBauswertung.Enabled = True
B_spektrenwahl.Enabled = True
Bid_key.Enabled = True
Bauto.Enabled = True
L_stat_auto.Text = "AUTO-Mode: OFF"
L_auto_endzeit.Text = ""
L_auto_ende.Visible = False
L_auto_endzeit.Visible = False
End Sub

Private Sub messen_fct(ByVal auto As Boolean)

```

```

Application.DoEvents()
Dim mess_cls As New messung

stat = "acq"

'entsprechende Teile des GUI aktivieren/deaktivieren
GBaktMessung.Enabled = True
GBparameter.Enabled = False
GBcalibrate.Enabled = False
GBanalyse.Enabled = False
GBauswertung.Enabled = False
B_spektrenwahl.Enabled = False
Bid_key.Enabled = False
Bauto.Enabled = False
L_anal_stat.Visible = False

mess_cls.messen(stat, laengenvorg)      'Aufruf der Messklasse
stat = ""
If (error_num = 0 And auto = False) Then System.Windows.Forms.MessageBox.Show _
    ("Messung erfolgreich beendet.")

'entsprechende Teile des GUI aktivieren/deaktivieren
GBcalibrate.Enabled = True
GBanalyse.Enabled = True
GBauswertung.Enabled = True
B_spektrenwahl.Enabled = True
Bid_key.Enabled = True
Bauto.Enabled = True
GBaktMessung.Enabled = False
End Sub

Private Sub kalibrieren_fct()
Application.DoEvents()
Dim anal_cls As New analyse

If (amount > 0) Then
'entsprechende Teile des GUI aktivieren/deaktivieren
stat = "cal"
GBparameter.Enabled = False
GBcalibrate.Enabled = False
GBanalyse.Enabled = False
GBauswertung.Enabled = False
B_spektrenwahl.Enabled = False
Bid_key.Enabled = False
Bauto.Enabled = False

anal_cls.kalibrieren(anal_files, stat)      'Aufruf der Analyseklasse

stat = ""
'entsprechende Teile des GUI aktivieren/deaktivieren
GBparameter.Enabled = True
GBcalibrate.Enabled = True
GBanalyse.Enabled = True
GBauswertung.Enabled = True
B_spektrenwahl.Enabled = True
Bid_key.Enabled = True
Bauto.Enabled = True

Else
System.Windows.Forms.MessageBox.Show("Kein Spektrum ausgewählt!")
End If
End Sub

Private Sub ipf_fct()
Application.DoEvents()
Dim anal_cls As New analyse

If (amount > 0) Then
Me.Enabled = False

anal_cls.ipf(anal_files)      'Aufruf Analyseklasse

Me.Enabled = True

Else
System.Windows.Forms.MessageBox.Show("Kein Spektrum ausgewählt!")
End If
End Sub

Private Sub analyse_std_fct()
Application.DoEvents()
Dim anal_cls As New analyse

If (amount > 0) Then
stat = "anal"

'entsprechende Teile des GUI aktivieren/deaktivieren
GBparameter.Enabled = False

```

```

GBcalibrate.Enabled = False
GBanalyse.Enabled = False
GBauswertung.Enabled = False
B_spektrenwahl.Enabled = False
Bid_key.Enabled = False
Bauto.Enabled = False

anal_cls.analyse_std(anal_files, stat)           'Aufruf Analyseklasse

stat = ""
If error_num = 0 Then System.Windows.Forms.MessageBox.Show _
    ("Analyse erfolgreich durchgeführt!")

'entsprechende Teile des GUI aktivieren/deaktivieren
GBparameter.Enabled = True
GBcalibrate.Enabled = True
GBanalyse.Enabled = True
GBauswertung.Enabled = True
B_spektrenwahl.Enabled = True
Bid_key.Enabled = True
Bauto.Enabled = True

Else
    System.Windows.Forms.MessageBox.Show("Kein Spektrum ausgewählt!")
End If
End Sub

Private Sub analyse_spez_fct(ByVal auto As Boolean)
    Application.DoEvents()
    Dim anal_cls As New analyse

    If (amount > 0) Then
        stat = "sanal"

        'entsprechende Teile des GUI aktivieren/deaktivieren
        GBparameter.Enabled = False
        GBcalibrate.Enabled = False
        GBanalyse.Enabled = False
        GBauswertung.Enabled = False
        B_spektrenwahl.Enabled = False
        Bid_key.Enabled = False
        Bauto.Enabled = False

        anal_cls.analyse_spez(anal_files, stat, auto)           'Aufruf Analyseklasse

        stat = ""
        If (error_num = 0 And auto = False) Then System.Windows.Forms.MessageBox.Show _
            ("Spezialanalyse erfolgreich durchgeführt!")

        'entsprechende Teile des GUI aktivieren/deaktivieren
        GBparameter.Enabled = True
        GBcalibrate.Enabled = True
        GBanalyse.Enabled = True
        GBauswertung.Enabled = True
        B_spektrenwahl.Enabled = True
        Bid_key.Enabled = True
        Bauto.Enabled = True

    Else
        System.Windows.Forms.MessageBox.Show("Kein Spektrum ausgewählt!")
    End If
End Sub

Private Sub auswertung_fct(ByVal auto As Boolean)
    Application.DoEvents()
    Dim eval_cls As New auswertung

    L_anal_stat.Text = "Auswertung läuft.."
    stat = "ausw"
    'entsprechende Teile des GUI aktivieren/deaktivieren
    GBparameter.Enabled = False
    GBcalibrate.Enabled = False
    GBanalyse.Enabled = False
    GBauswertung.Enabled = False
    B_spektrenwahl.Enabled = False
    Bid_key.Enabled = False
    Bauto.Enabled = False

    eval_cls.auswerten(stat)           'Aufruf Auswertungsklasse

    stat = ""
    If (amount = 1) Then
        L_anal_stat.Text = amount & " Spektrum ausgewählt"
    ElseIf (amount > 1) Then
        L_anal_stat.Text = amount & " Spektren ausgewählt"
    Else
        L_anal_stat.Text = " Keine Spektren ausgewählt"
    End If
End If

```

```

If (error_num = 0 And auto = False) Then System.Windows.Forms.MessageBox.Show _
("Auswertung erfolgreich durchgeführt!")

'entsprechende Teile des GUI aktivieren/deaktivieren
GBparameter.Enabled = True
GBcalibrate.Enabled = True
GBanalyse.Enabled = True
GBauswertung.Enabled = True
B_spektrenwahl.Enabled = True
Bid_key.Enabled = True
Bauto.Enabled = True
End Sub
'-----

'ALLGEMEINE FUNKTIONEN
Private Sub Bid_key_Click(sender As Object, e As EventArgs) Handles Bid_key.Click
'ID-Referenz von ausgewählten Spektren anzeigen lassen

Dim ref_msg As String

If (amount > 0) Then
For j = 0 To (amount - 1)
camdata.Open(anal_files(j), 512) 'Spektrum öffnen
ref_msg = "Dateiname: " & System.IO.Path.GetFileName(anal_files(j)) _
& vbCrLf & "Referenz: " & camdata.Param(536936537) _
& camdata.Param(536936540) 'Ausgabe erstellen / Referenz auslesen

System.Windows.Forms.MessageBox.Show(ref_msg) 'Referenz anzeigen
camdata.Close(0)
Next j
Else
System.Windows.Forms.MessageBox.Show("Kein Spektrum ausgewählt!")
End If
End Sub

Private Sub abbruch()
'Abbrechen der aktuellen Durchführung

If (stat = "acq") Then 'Während Messung
error_num = 1 'Fehlerflag setzen
Shell("STOPMCA det:DLFC /Abort") 'Messung stoppen
Lprobe_nr.Text = "0" 'Anzeige Probennr. zurücksetzen
Ldauer_tim.Text = "0" 'Anzeige Messdauer zurücksetzen
Ldate.Visible = False
System.Windows.Forms.MessageBox.Show("Messung wurde abgebrochen!" & vbCrLf &
"Daten sind nicht vollständig - Spektren nicht verwenden!")
L_mess_stat.Text = "Abbrechen.."

Elseif (stat = "anal") Then 'Während Standardanalyse
error_num = 1
System.Windows.Forms.MessageBox.Show("Analyse wurde abgebrochen!" & vbCrLf &
"Daten sind nicht vollständig - Reportfiles nicht verwenden!")
L_anal_stat.Text = "Abbrechen.."

Elseif (stat = "sanal") Then 'Während Spezialanalyse
error_num = 1
System.Windows.Forms.MessageBox.Show("Spezielle Analyse wurde abgebrochen!" & vbCrLf &
"Daten sind nicht vollständig - Reportfiles nicht verwenden!")
L_anal_stat.Text = "Abbrechen.."

Elseif (stat = "cal") Then 'Während Kalibrierung
error_num = 1
System.Windows.Forms.MessageBox.Show("Kalibrierung wurde abgebrochen!" & vbCrLf &
"Daten sind nicht vollständig - Spektren nicht verwenden!")
L_anal_stat.Text = "Abbrechen.."

Elseif (stat = "ausw") Then 'Während Auswertung
error_num = 1
System.Windows.Forms.MessageBox.Show("Auswertung wurde abgebrochen!" & vbCrLf &
"Daten und Logfile möglicherweise nicht vollständig.")
L_anal_stat.Text = "Abbrechen.."

End If
End Sub

Private Sub Bbeenden_Click(sender As Object, e As EventArgs) Handles Bbeenden.Click
'Programm beenden

Dim control As New control_box

control.Text = "Beenden"
control.Ltext.Text = "Soll wirklich abgebrochen und das Programm beendet werden?"

If control.ShowDialog() = Windows.Forms.DialogResult.OK Then

If (stat = "acq" Or stat = "anal" Or stat = "sanal" _
Or stat = "cal" Or stat = "ausw") Then
abbruch() 'Abbruch aufrufen wenn Sequenz aktiv

```

```

        End If
        Me.Close()
    End If

End Sub
'-----

'ÖFFENTLICHE FUNKTIONEN

Public Sub sleep_fct(warten As Integer)
'Delay Funktion zum Stoppen des Codes für beliebige Anzahl an Sekunden

    Dim endzeit As DateTime
    Ldauer_tim.Visible = True
    endzeit = DateAdd(DateInterval.Second, warten, Now)

    Do While DateTime.Now < endzeit
        Application.DoEvents()
        If error_num = 1 Then Exit Do

        Ldauer_tim.Text = (endzeit.Subtract(DateTime.Now)).ToString("dd\.\hh\:\mm\:\ss")
    Loop
    Ldauer_tim.Visible = False
End Sub

Public Sub Update_fct(status As String, y As Integer)
'Für Statusänderungen bzw. Fortschrittsleistenupdate aus anderen Klassen

    Application.DoEvents()
    L_anal_stat.Visible = True

    If (status = "anal") Then
        L_anal_stat.Text = "Analyse läuft.."
    ElseIf (status = "sanal") Then
        L_anal_stat.Text = "Spezialanalyse läuft.."
        If (y = 1) Then PBprogress.PerformStep()
    ElseIf (status = "cal") Then
        L_anal_stat.Text = "Kalibrierung läuft.."
    ElseIf (status = "ausw") Then
        L_anal_stat.Text = "Auswertung läuft.."
        If (y = 1) Then PBprogress.PerformStep()
    ElseIf (status = "logf") Then
        L_anal_stat.Text = "Logfile erstellen"
        If (y = 1) Then PBprogress.PerformStep()
    ElseIf (status = "") Then
        L_anal_stat.Text = ""
        L_anal_stat.Visible = False
    End If
End Sub
'-----

'Unterbinden von Sonderzeichen und Falscheingaben bei Textfeldern

Private Sub TBanzahl_KeyDown(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles TBanzahl_KeyDown
'bei KEY-DOWN EVENT werden manche Zeichen erlaubt, der Rest unterdrückt

    Select Case e.KeyCode
        Case Keys.D0 To Keys.D9, Keys.Back, Keys.Decimal, Keys.Delete,
            Keys.Left, Keys.Right, Keys.NumPad0 To Keys.NumPad9
            'Zahlen(Zeile u. Numpad), Backspace, Links, Rechts und Entfernen zulassen

            If e.Modifiers = Keys.Shift Then
                e.SuppressKeyPress = True
            ElseIf e.Modifiers = 393216 Then
                e.SuppressKeyPress = True
            End If

        Case Else
            e.SuppressKeyPress = True ' alle anderen Eingaben unterdrücken
    End Select
End Sub

Private Sub TBmessdauer_KeyDown(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles TBmessdauer_KeyDown
'siehe oben

    Select Case e.KeyCode
        Case Keys.D0 To Keys.D9, Keys.Back, Keys.Decimal, Keys.Delete,
            Keys.Left, Keys.Right, Keys.NumPad0 To Keys.NumPad9
            'Zahlen(Zeile u. Numpad), Backspace, Links, Rechts und Entfernen zulassen

            If e.Modifiers = Keys.Shift Then
                e.SuppressKeyPress = True
            ElseIf e.Modifiers = 393216 Then
                e.SuppressKeyPress = True
            End If
    End If
End Sub

```

```

        Case Else
            e.SuppressKeyPress = True ' alle anderen Eingaben unterdrücken
        End Select
    End Sub

Private Sub TBdateiname_KeyDown(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles
TBdateiname.KeyDown
    'siehe oben

    If e.Modifiers = Keys.Shift Then 'Shift
        e.SuppressKeyPress = True
    ElseIf e.Modifiers = 393216 Then 'Alt, Strg, Alt Gr
        e.SuppressKeyPress = True
    ElseIf e.KeyCode = Keys.Space Then 'Space
        e.SuppressKeyPress = True
    End If
End Sub

Private Sub TBenergie_KeyDown(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles TBenergie.KeyDown
    'siehe oben

    Select Case e.KeyCode
        Case Keys.D0 To Keys.D9, Keys.Back, Keys.Decimal, Keys.Delete,
            Keys.Left, Keys.Right, Keys.NumPad0 To Keys.NumPad9
            ' Zahlen(Zeile u. Numpad), Backspace, Links, Rechts und Entfernen zulassen

            If e.Modifiers = Keys.Shift Then
                e.SuppressKeyPress = True
            ElseIf e.Modifiers = 393216 Then
                e.SuppressKeyPress = True
            End If

        Case Else
            e.SuppressKeyPress = True ' alle anderen Eingaben unterdrücken
        End Select
    End Sub
End Sub
'-----
'SPEKTRENAUSWAHL / ORDNERAUSWAHL / DATEIENAUSWAHL
Private Sub RBmittel_acq_CheckedChanged(sender As Object, e As EventArgs) Handles RBmittel_acq.CheckedChanged
    TBanzahl.Text = "50"
    TBmessdauer.Text = "1800"
    laengenvorg = "mittel"
End Sub

Private Sub RBlang_acq_CheckedChanged(sender As Object, e As EventArgs) Handles RBlang_acq.CheckedChanged
    TBanzahl.Text = "50"
    TBmessdauer.Text = "10000"
    laengenvorg = "lang"
End Sub

Private Sub Bpfad_Click(sender As Object, e As EventArgs) Handles Bpfad_mess.Click
    'Speicherort für Messspektren auswählen
    If FBD_mess.ShowDialog() = DialogResult.OK Then 'Folder Browser Dialog
        TBpfad.Text = FBD_mess.SelectedPath & "\"
    End If
End Sub

Private Sub Bspektrenwahl_Click(sender As Object, e As EventArgs) Handles B_spektrenwahl.Click
    'Spektren für Analyse/Kalibrierung/IPF auswählen

    If OFD_cnf.ShowDialog() = DialogResult.OK Then 'Folder Browser Dialog
        anal_files = OFD_cnf.FileNames
        amount = anal_files.Count

        If (amount = 1) Then
            L_anal_stat.Text = amount & " Spektrum ausgewählt"
        Else
            L_anal_stat.Text = amount & " Spektren ausgewählt"
        End If

    ElseIf (DialogResult.Cancel) Then
        amount = 0
        L_anal_stat.Text = "Kein Spektrum ausgewählt"
    End If
    L_anal_stat.Visible = True
End Sub

Private Sub Bcalpath_Click(sender As Object, e As EventArgs) Handles Bcalpath.Click
    'Ordnerpfad für Kalibrierdatei auswählen
    If OFD_cal.ShowDialog() = DialogResult.OK Then 'Folder Browser Dialog
        TBcalpath.Text = OFD_cal.FileName
    End If
End Sub

Private Sub B_asf_wahl_Click(sender As Object, e As EventArgs) Handles B_asf_wahl.Click
    'Ordnerpfad für Analyse - Analysesequenz (5 SIGMA_OR) auswählen

```

```

    If OFD_asf.ShowDialog() = DialogResult.OK Then
        TB_asf.Text = System.IO.Path.GetFileName(OFD_asf.FileName)
    End If
End Sub

Private Sub Btxt_wahl_Click(sender As Object, e As EventArgs) Handles Btxt_wahl.Click
    'Ordnepfad für Analyse - peaks.txt auswählen
    If OFD_txt.ShowDialog() = DialogResult.OK Then
        TBtxt.Text = OFD_txt.FileName
    End If
End Sub

Private Sub Bpfad_ausw_Click(sender As Object, e As EventArgs) Handles Bpfad_ausw.Click
    'Ordnepfad für Auswertung auswählen
    If FBD_ausw.ShowDialog() = DialogResult.OK Then
        TB_ordner_ausw.Text = FBD_ausw.SelectedPath & "\"
    End If
End Sub
'-----

'BUTTON-CLICK-EVENTS für Funktionsweiterleitung
Private Sub Bmessen_Click(sender As Object, e As EventArgs) Handles Bmessen.Click
    Call messen_fct(False)
    'AUTO durchführung = false
End Sub
Private Sub Bpfp_Click(sender As Object, e As EventArgs) Handles Bpfp.Click
    Call ipf_fct()
End Sub
Private Sub Bkalibrieren_Click(sender As Object, e As EventArgs) Handles Bkalibrieren.Click
    Call kalibrieren_fct()
End Sub
Private Sub Bauto_Click(sender As Object, e As EventArgs) Handles Bauto.Click
    Call auto_fct()
End Sub
Private Sub Banalyse_Click(sender As Object, e As EventArgs) Handles Banalyse.Click
    Call analyse_std_fct()
End Sub
Private Sub Bspez_anal_Click(sender As Object, e As EventArgs) Handles Bspez_anal.Click
    Call analyse_spez_fct(False)
    'AUTO durchführung = false
End Sub
Public Sub Bevaluation_Click(sender As Object, e As EventArgs) Handles Bevaluation.Click
    Call auswertung_fct(False)
    'AUTO durchführung = false
End Sub
Private Sub Babbruch_Click(sender As Object, e As EventArgs) Handles Babbruch.Click
    Dim control As New control_box

    control.Text = "Abbruch"
    control.Ltext.Text = "Soll wirklich abgebrochen werden?"

    If control.ShowDialog() = Windows.Forms.DialogResult.OK Then
        abbruch()
    End If
End Sub
End Class

```

## TOCHTERKLASSE - AUTO\_AUSWAHL.VB

```

Public Class auto_auswahl
    '{32 Zeilen}
    'Auswahl der Parameter für automatische Messung/Analyse/Auswertung - Sequenz durchführung

    Public Function auto_ausw_fct(ByRef mit_checked As Boolean, ByRef std_choice As Boolean)
        'mit_checked: Mittlere Dateien können analysiert werden, wenn Reportfiles noch nicht vorhanden

        'std_choice = 0 für manuelle Auswahl der Standards -FALSE
        'std_choice = 1 für automatische Auswahl der Standards -TRUE

        std_choice = False
        If Me.ShowDialog() = Windows.Forms.DialogResult.OK Then
            mit_checked = CB_auto_ch_mit.CheckState
            'Form wird als Dialog angezeigt
            'Auswahl für Analyse der mittleren Spektren
            If RBauto.Checked Then std_choice = True
            'Auswahl für automatische Überprüfung der Standards
            Return 0
        End If
        Else
            'Abbruch
            Return 1
        End If
    End Function

    Private Sub Bgo_Click(sender As Object, e As EventArgs) Handles Bgo.Click
        Me.DialogResult = System.Windows.Forms.DialogResult.OK
        Me.Close()
        'positives Dialog Result
    End Sub

```

```

Private Sub Bexit_Click(sender As Object, e As EventArgs) Handles Bexit.Click
    Me.DialogResult = System.Windows.Forms.DialogResult.Cancel      'negatives Dialog Result
    Me.Close()
End Sub
End Class

```

## TOCHTERKLASSE - CONTROL\_BOX.VB

```

Public Class control_box      '{13 Zeilen}
    'Kontrollabfrage für Abbruch/Beenden

    Private Sub Byes_Click(sender As Object, e As EventArgs) Handles Byes.Click
        Me.DialogResult = System.Windows.Forms.DialogResult.OK      'positives Dialog Result
        Me.Close()
    End Sub

    Private Sub Bno_Click(sender As Object, e As EventArgs) Handles Bno.Click
        Me.DialogResult = System.Windows.Forms.DialogResult.Cancel  'negatives Dialog Result
        Me.Close()
    End Sub
End Class

```

## KLASSE - MESSUNG.VB

```

Public Class messung      '{306 Zeilen}
    Dim i, j, k As Integer
    Dim id_ref1 As String      'Reproduzierbare Arbeitsdurchführung-Referenz - Teil 1
    Dim id_ref2 As String      'Reproduzierbare Arbeitsdurchführung-Referenz - Teil 2

    Dim camdata As New CanberraDataAccessLib.DataAccess      'neue Instanz der Canberra-DataAccess-Klasse

    Public Sub messen(ByVal stat As String, vorg_length As String)
        Application.DoEvents()

        DLFC_index.error_num = 0      'Fehlervariable initialisieren

        'Messvariablen
        Dim anzahl As Integer      'Probenanzahl (Anwender)
        Dim messdauer As Integer      'Messdauer (Anwender)
        Dim pfad As String      'Pfad (Anwender)
        Dim dateiname As String      'Dateiname (Anwender)
        Dim filename As String      'Dateiname in der Schleife
        Dim filepath As String      'Dateipfad zusammengesetzt aus Pfad, Dateiname
        Dim date_var As Integer      'Var zum berechnen des Datums

        Dim index As Integer      'Hauptschleifenindex

        'Parameter der CAM Datei
        Dim spec_size As Integer      'Anzahl der Kanäle des Spektrum
        Dim name_UNC As String      'Name des unkorrigierten Spektrum
        Dim name_LFC As String      'Name des korrigierten Spektrum
        Dim integral_corr As Decimal      'korrigiertes Integral
        Dim integral_uncorr As Decimal      'unkorrigiertes Integral
        Dim elapsedtime_true As Double      'vergangene echte Zeit
        Dim elapsedtime_live As Double      'vergangene Messzeit
        Dim elapsedtime_live_corr As Double      'vergangene Messzeit
        Dim ICR As Single      'Incoming Count Rate

        Dim data() As Single      'Datenarray des gesamten Spektrum
        Dim data_UNC(8191) As Single      'Datenarray des unkorrigiertes Spektrum
        Dim data_LFC(8191) As Single      'Datenarray des korrigiertes Spektrum

        '-----
        '-----Variablen initialisieren-----
        stat = "acq"
        spec_size = 0
        name_UNC = ""
        name_LFC = ""
        integral_corr = 0
        integral_uncorr = 0
        elapsedtime_live = 0
        elapsedtime_live_corr = 0
        elapsedtime_true = 0
        ICR = 0

        '---Überprüfe eingaben---
        If (DLFC_index.TBanzahl.Text = "" Or DLFC_index.TBmessdauer.Text = "" Or

```

```

DLFC_index.TBpfad.Text = "" Or DLFC_index.TBdateiname.Text = "" Then

System.Windows.Forms.MessageBox.Show("Fehlende Parameter!")
DLFC_index.error_num = 1
GoTo LastLine_acq

Elseif (Not System.IO.Directory.Exists(DLFC_index.TBpfad.Text)) Then
System.Windows.Forms.MessageBox.Show("Speicherort existiert nicht!")
DLFC_index.error_num = 1
GoTo LastLine_acq
End If

'-----Daten einlesen / Ordnerstruktur erstellen-----
anzahl = CType(DLFC_index.TBanzahl.Text, Integer)           'Probenanzahl einlesen
messdauer = CType(DLFC_index.TBmessdauer.Text, Integer)   'Messdauer einlesen
pfad = DLFC_index.TBpfad.Text
dateiname = DLFC_index.TBdateiname.Text                   'Dateiname einlesen
filepath = ""

Call Create_directory_struct(pfad, dateiname, vorg_length)

'Benutzeroberfläche - laufende Messdaten
DLFC_index.Lprobe_nr.Text = "0"                            'Anzeige Probennr. initialisieren
DLFC_index.Ldauer_tim.Text = "0"                          'Anzeige Messdauer initialisieren
date_var = (anzahl * (messdauer + 30 + 5))                 'ACHTUNG: anpassen wenn Delay verändert wird (5 aufgrund Laufzeittests)
DLFC_index.Ldate.Text = DateAdd(DateInterval.Second, date_var, Now).
ToString("dd.MM.yyyy HH:mm")                             'Messungsende anzeigen
DLFC_index.Lprobe_nr.Visible = True                       'Label anzeigen
DLFC_index.Ldauer_tim.Visible = True                      'Label anzeigen
DLFC_index.Ldate.Visible = True                          'Label anzeigen

DLFC_index.error_num = check_files(anzahl, pfad, dateiname) 'Überprüfung ob Spektren bereits vorhanden
If DLFC_index.error_num = 1 Then GoTo LastLine_acq

'-----SCHLEIFE-----
'laufzeittest = DateTime.Now                               'Laufzeittest des Messung

index = 1                                                 'Index initialisieren
Do While index <= anzahl

DLFC_index.L_mess_stat.Text = "Messung läuft"
DLFC_index.Ldauer_tim.Text = "0"
DLFC_index.Lprobe_nr.Text = CType(index, String)         'Anzeige Messdauer zurücksetzen
filename = dateiname                                     'Anzeige Probennr. setzen
Dateiname initialisieren

If index < 10 Then
filename = filename & "_000" & CType(index, String)     'IF für Probenanzahl bis 9999
Elseif index > 9 & index < 100 Then
filename = filename & "_00" & CType(index, String)
Elseif index > 99 & index < 1000 Then
filename = filename & "_0" & CType(index, String)
Elseif index > 999 Then
filename = filename & "_" & CType(index, String)
End If

Shell("STARTMCA det:DLFC /realpreset=" & CType(messdauer, String), vbHide, True) 'Messung starten - mit Variable funktioniert!
'Genie: Messung abwarten
Call DLFC_index.sleep_fct(messdauer)                    'Programm wartet Messdauer ab
DLFC_index.Ldauer_tim.Text = CType(Val(messdauer), String) 'Anzeige Messdauer auf Endzeit setzen

If DLFC_index.error_num = 1 Then GoTo LastLine_acq      'Error: Abbruch

'LFC & UNC als DUAL Spektrum speichern
filepath = "" & pfad & "raw\" & filename & "_DUAL.CNF"" 'Dateipfad modifizieren für Speicherbefehl
Shell("MOVEDATA det:DLFC " & filepath, vbHide, True)   'DUAL Spektrum am angegebenen Speicherort speichern,
True damit Shell befehl abgewartet wird

'LFC Datei erstellen
name_LFC = filename & "_LFC.CNF"
filepath = "" & pfad & name_LFC & ""
Shell("MOVEDATA det:DLFC " & filepath, vbHide, True)   'Dateiname modifizieren
True damit Shell befehl abgewartet wird
Dateipfad modifizieren für Speicherbefehl
'LFC Spektrum am angegebenen Speicherort speichern,

'UNC Datei erstellen
name UNC = filename & "_UNC.CNF"
filepath = "" & pfad & name UNC & ""
Shell("MOVEDATA det:DLFC " & filepath, vbHide, True)   'Dateiname modifizieren
True damit Shell befehl abgewartet wird
Dateipfad modifizieren für Speicherbefehl
'UNC Spektrum am angegebenen Speicherort speichern,

'Daten auslesen
filepath = pfad & "raw\" & filename & "_DUAL.CNF"     'Pfad des Spektrum definieren
camdata.Open(filepath, 512)                             'Öffnen des Spektrum
data = camdata.Spectrum()                               'Daten in Variable übertragen
spec_size = camdata.Param(536870919)                   'Spektrumgröße übertragen
elapsedtime_true = camdata.Param(536870934)             'CAM_X_EREAL- Messzeit (tatsächlich vergangene Zeit)

```

```

elapsedtime_live = camdata.Param(536870935)          'CAM_X_ELIVE - Detektionszeit (Zeit in welcher Ereignisse detektiert werden -
Messzeit-Totzeit)

'Referenz erstellen
id_ref1 = "Q" & DateTime.Now.ToString("MMyy") & "dua/"  'Erstellen
id_ref2 = ""
camdata.Param(536936537) = id_ref1                    'Eintragen
camdata.Param(536936540) = id_ref2
-----

camdata.Save("")
camdata.Close()                                     'Duales Spektrum schließen

'--- UNC Spektrum erstellen ---
integral_uncorr = 0
j = 0
For j = 0 To ((spec_size / 2) - 1)
    data UNC(j) = data(j + (spec_size / 2))            'UNC Spektrum erstellen (halbes DUAL)
    integral_uncorr = integral_uncorr + data UNC(j)    'Summe aller Counts aller Kanäle des UNC Spektrum
Next j

'--- LFC Spektrum erstellen ---
integral_corr = 0
j = 0
For j = 0 To ((spec_size / 2) - 1)
    data_LFC(j) = data(j)                             'LFC Spektrum erstellen (halbes DUAL)
    integral_corr = integral_corr + data_LFC(j)        'Summe aller Counts aller Kanäle des LFC Spektrum
Next j

If (integral_corr > 0 And elapsedtime_live > 0) Then
    elapsedtime_live_corr = elapsedtime_true * (integral_uncorr / integral_corr)  'korrigierte Detektionszeit
    ICR = integral_uncorr / elapsedtime_live                                     'Incoming Count Rate
End If

'----- UNC.CNF Datei erstellen -----
id_ref1 = "Q" & DateTime.Now.ToString("MMyy") & "unc/"  'Referenz erstellen

camdata.Open(pfad & name UNC, 512)                    '512 - read and write
camdata.Spectrum(1, (spec_size / 2)) = data UNC      'Spektrumdaten übertagen
camdata.Param(536870919) = spec_size / 2            'Spektrumgröße definieren
camdata.Param(536870934) = elapsedtime_true         'Messzeit eintragen
camdata.Param(536870935) = elapsedtime_live_corr    'Detektionszeit eintragen
camdata.Param(536936537) = id_ref1                  'CAM_T_SDESC1: Beschreibungen
camdata.Param(536936538) = "DLFC uncorrected"      'CAM_T_SDESC2
camdata.Param(536936539) = "ICR =" & Str(ICR) & " cps" 'CAM_T_SDESC3
camdata.Param(536936540) = id_ref2                  'CAM_T_SDESC4
camdata.Param(536871094) = 0                        'CAM_L_ASTFBADCAL = 0: keine fehlerhaften Kalibrierungsdaten
camdata.Save("")                                    'Flush CAM buffer
camdata.Close()

'----- LFC.CNF Datei erstellen -----
id_ref1 = "Q" & DateTime.Now.ToString("MMyy") & "lfc/"  'Referenz erstellen

camdata.Open(pfad & name_LFC, 512)                    '512 - read and write
camdata.Spectrum(1, spec_size / 2) = data_LFC      'Spektrumdaten übertagen
camdata.Param(536870919) = spec_size / 2            'Spektrumgröße definieren
camdata.Param(536870934) = elapsedtime_true         'Messzeit eintragen
camdata.Param(536870935) = elapsedtime_true         'Detektionszeit eintragen
camdata.Param(536936537) = id_ref1                  'CAM_T_SDESC1: Beschreibungen
camdata.Param(536936538) = "DLFC corrected"        'CAM_T_SDESC2
camdata.Param(536936539) = "ICR =" & Str(ICR) & " cps" 'CAM_T_SDESC3
camdata.Param(536936540) = id_ref2                  'CAM_T_SDESC4
camdata.Param(536871094) = 0                        'CAM_L_ASTFBADCAL = 0: keine fehlerhaften Kalibrierungsdaten
camdata.Save("")                                    'Flush CAM buffer
camdata.Close()

If DLFC_index.error_num = 1 Then GoTo LastLine_acq    'Error: Abbruch

'----- Probenwechsel -----
If anzahl > 1 Then
    Call change_sample()
    If DLFC_index.error_num = 1 Then GoTo LastLine_acq 'Error: Abbruch
End If

index += 1                                          'Schleifenindex erhöhen
Loop

LastLine_acq:

'Anzeigen zurücksetzen und Statusfelder verbergen
DLFC_index.Lprobe_nr.Text = "0"
DLFC_index.Ldauer_tim.Text = "0"
DLFC_index.Ldate.Text = ""
DLFC_index.L_mess_stat.Visible = False
DLFC_index.Lprobe_nr.Visible = False
stat = ""
DLFC_index.GBparameter.Enabled = True
End Sub

```

```

Private Sub change_sample()
'----- Probenwechsel -----

Dim startinfo As New ProcessStartInfo("C:\Program Files\AutoMeasure_DLFC\DLFC_ADV.exe") 'Probenwechsler EXE
startinfo.WindowStyle = ProcessWindowStyle.Hidden 'Probenwechsler EXE versteckt ausführen

DLFC_index.L_mess_stat.Text = "Probenwechsel"
DLFC_index.L_mess_stat.ForeColor = Color.Blue 'Schriftfarbe der Anzeige ändern
DLFC_index.L_mess_stat.Visible = True 'Anzeige Probenwechsel anzeigen
Process.Start(startinfo) 'Probenwechsler starten
DLFC_index.sleep_fct(30) 'Programmstop während Probenwechsel
End Sub

Private Sub Create_directory_struct(ByRef _pfad As String, ByRef _dateiname As String, _vorg_length As String)

'Root-Ordner überprüfen --> ansonsten erstellen
If (Not (System.IO.Directory.Exists(_pfad & _dateiname))) Then
My.Computer.FileSystem.CreateDirectory(_pfad & _dateiname)
End If

'-----Pfad einlesen - mit optionaler Messungs-Länge -----
If (DLFC_index.RBmittel_acq.Checked = True Or DLFC_index.RBlang_acq.Checked = True) Then
_pfad = DLFC_index.TBpfad.Text & _dateiname & "\" & _vorg_length & "\"
If (Not (System.IO.Directory.Exists(_pfad))) Then 'Ordner überprüfen --> ansonsten erstellen
My.Computer.FileSystem.CreateDirectory(_pfad)
End If
_dateiname = DLFC_index.TBdateiname.Text & "_" & _vorg_length 'By Reference Dateiname ändern
Else
_pfad = DLFC_index.TBpfad.Text & _dateiname & "\" 'By Reference Dateipfad ändern
End If

If (Not (System.IO.Directory.Exists(_pfad & "raw"))) Then 'Ordner überprüfen --> ansonsten erstellen
My.Computer.FileSystem.CreateDirectory(_pfad & "raw")
End If

End Sub

Private Function check_files(_anzahl As Integer, _pfad As String, _dateiname As String)

'SCHLEIFE zum Abfragen ob eine der Zieldateien bereits vorhanden ist
For j = 1 To _anzahl
Dim _filename = _dateiname
If j < 10 Then
_filename = _filename & "_000" & CType(j, String) 'IF für Probenanzahl bis 9999
Elseif j > 9 & j < 100 Then
_filename = _filename & "_00" & CType(j, String)
Elseif j > 99 & j < 1000 Then
_filename = _filename & "_0" & CType(j, String)
Elseif j > 999 Then
_filename = _filename & "_" & CType(j, String)
End If

If (System.IO.File.Exists(_pfad & "raw\" & _filename & "_DUAL.CNF")) Then 'Datei existiert bereits --> Fehler
System.Windows.Forms.MessageBox.Show _
("Datei ist im Zielordner bereits vorhanden! Messabbruch." _
& vbCrLf & _pfad & _filename & "_DUAL.CNF")
Return 1
End If

If (System.IO.File.Exists(_pfad & _filename & "_LFC.CNF")) Then 'Datei existiert bereits --> Fehler
System.Windows.Forms.MessageBox.Show _
("Datei ist im Zielordner bereits vorhanden! Messabbruch." _
& vbCrLf & _pfad & _filename & "_LFC.CNF")
Return 1
End If

If (System.IO.File.Exists(_pfad & _filename & "_UNC.CNF")) Then 'Datei existiert bereits --> Fehler
System.Windows.Forms.MessageBox.Show _
("Datei ist im Zielordner bereits vorhanden! Messabbruch." _
& vbCrLf & _pfad & _filename & "_DUAL.CNF")
Return 1
End If
Next
Return 0
End Function
End Class

```

## KLASSE - ANALYSE.VB

```

Public Class analyse {539 Zeilen}
'Variablen, welche alle Funktionen der Klasse verwenden
Dim i, j, k, l, m As Integer 'Schleifenindizes --> l, m hinzugefügt

```

```

Dim amount As Integer                'Zahl der ausgewählten Spektren
Dim id_ref1 As String                'Reproduzierbarer Weg-Referenz, Teil 1
Dim id_ref2 As String                'Reproduzierbarer Weg-Referenz, Teil 2
Dim refADD As String                 'hinzuzufügende Referenz

Dim camdata As New CanberraDataAccessLib.DataAccess    'neue Instanz der Canberra-DataAccess-Klasse
Dim anal As New CanberraSequenceAnalyzerLib.SequenceAnalyzer    'neue Instanz der Canberra-SequenceAnalyzer-Klasse
Dim report_anal As New CanberraReporterLib.ReportViewer    'neue Instanz der Canberra-Reporter-Klasse

Public Sub kalibrieren(files() As String, ByVal stat As String)
    'Kalibrierfunktion für Übertragung von Kalibrierungen von Kalibrierdatei auf Spektren

    Application.DoEvents()            'Windows-Events von Form bearbeiten
    Dim calfile As String              'Kalibrierdatei
    Dim check As Integer = 0           'Kontrollvariable

    amount = files.Count               'Spektrenanzahl setzen
    stat = "cal"                       'Status setzen
    DLFC_index.error_num = 0           'Fehlervariable initialisieren
    Call DLFC_index.Update_fct("cal", 0) 'Update_fct aufrufen

    calfile = DLFC_index.TBcalpath.Text 'Kalibrierdatei übertragen
    DLFC_index.PBprogress.Value = 0    'Fortschrittsbalken initialisieren
    DLFC_index.PBprogress.Step = CInt(100 / amount) 'Fortschrittsbalken-Schrittgröße definieren

    'Kalibrierschleife
    For j = 0 To (amount - 1)
        If (calfile IsNot "" And System.IO.File.Exists(calfile)) Then
            If (DLFC_index.CBenergy.Checked And DLFC_index.CBeff.Checked) Then

                'Übertrage Energy&Effizienz - Kalibrierung von Kalibrierdatei auf Spektren
                Shell("MOVEDATA " & """" & calfile & """" & " " & """" & files(j) & """" _
                    & " /ECAL /EFFCAL /OVERWRITE", vbHide, True)
                check = 1

            ElseIf (DLFC_index.CBenergy.Checked And DLFC_index.CBeff.CheckState = CheckState.Unchecked) Then

                'Übertrage nur Energy - Kalibrierung von Kalibrierdatei auf Spektren
                Shell("MOVEDATA " & """" & calfile & """" & " " & """" & files(j) & """" _
                    & " /ECAL /OVERWRITE", vbHide, True)
                check = 1

            ElseIf (DLFC_index.CBenergy.CheckState = CheckState.Unchecked And DLFC_index.CBeff.Checked) Then

                'Übertrage nur Effizienz - Kalibrierung von Kalibrierdatei auf Spektren
                Shell("MOVEDATA " & """" & calfile & """" & " " & """" & files(j) & """" _
                    & " /EFFCAL /OVERWRITE", vbHide, True)
                check = 1

            Else
                System.Windows.Forms.MessageBox.Show("Keine Kalibrierung ausgewählt.")
                Exit For
            End If

            If (DLFC_index.error_num = 1) Then GoTo LastLine_cal
            DLFC_index.PBprogress.PerformStep()

            'Referenzierung der Kalibrierung
            refADD = "C" & System.IO.Path.GetFileNameWithoutExtension(calfile) 'Erstellung der Referenz
            Call referenz(files(j), refADD) '1,1 sind irrelevante übergabewerte

            Else
                System.Windows.Forms.MessageBox.Show("Korrekte Kalibrierdatei (.CAL oder .CNF) auswählen!")
            End If
        End If
    Next j

LastLine_cal:
    If (amount = 1) Then
        DLFC_index.L_anal_stat.Text = amount & " Spektrum ausgewählt"
    Else
        DLFC_index.L_anal_stat.Text = amount & " Spektren ausgewählt"
    End If
    stat = ""
    If (DLFC_index.error_num = 0 And check = 1) Then 'Erfolgsmeldung nur bei Kontrollvariable = 1
        System.Windows.Forms.MessageBox.Show("Kalibrierung durchgeführt.")
    End If
    DLFC_index.PBprogress.Value = 0
End Sub

Public Sub ipf(files() As String)
    'Interactive-Peak-Fit zur manuellen Bearbeitung oder Kontrolle von Spektren

    Application.DoEvents()            'Windows-Events von Form bearbeiten
    DLFC_index.error_num = 0           'Fehlervariable initialisieren
    Dim energie_wahl As Double         'Energie Filter
    Dim userADD As String = ""         'individuelle Benutzereingabe initialisieren
    Dim input As New input_box()       'Klasse Inputbox instanzieren

```

```

'IPF- Schleife
For j = 0 To files.Count - 1
  If (DLFC_index.TBenergie.Text <> "") Then
    energie_wahl = CType(DLFC_index.TBenergie.Text, Double)

    'IPF inklusive Filter aufrufen
    Shell("IPFIT " & """" & files(j) & """" & " /ENERGY=" & energie_wahl, vbHide, True)

  Elseif (DLFC_index.TBenergie.Text = "") Then

    'IPF ohne Filter aufrufen
    Shell("IPFIT " & """" & files(j) & """" & " /NOFILTERS", vbHide, True)

  End If

  userADD = input.eingabe(files(j))
  refADD = "ipf" & userADD          'Referenz erstellen
  Call referenz(files(j), refADD)   '1/1 als irrelevante übergabewerte
Next
End Sub

Public Sub analyse_std(anal_filenames() As String, ByVal stat As String)
'Standardanalyse zur Analyse von Spektren mittels Analysesequenz

Application.DoEvents()           'Windows-Events von Form bearbeiten

Dim asf_file As String           'Analysesequenzdatei
Dim rep_path As String           'Ausgabepfad für Reportdatei
DLFC_index.error_num = 0        'Fehlervariable initialisieren
amount = anal_filenames.Count    'Spektrenanzahl definieren

Call DUAL_DLFC.DLFC_index.Update_fct("anal", 0)    'Update_fct aufrufen
asf_file = DUAL_DLFC.DLFC_index.TB_asf.Text        'Analysesequenzdatei übertragen
DUAL_DLFC.DLFC_index.PBprogress.Value = 0         'Fortschrittsbalken initialisieren
DUAL_DLFC.DLFC_index.PBprogress.Step = CInt(100 / amount) 'Fortschrittsbalken-Schrittgröße definieren

If (System.IO.File.Exists("C:\GENIE2K\CTLFILERS" & asf_file)) Then

  'Überprüfung und Erstellung der Ordnerstruktur
  If (Not (System.IO.Directory.Exists(System.IO.Path.GetDirectoryName(anal_filenames(0)) _
    & "\reports"))) Then
    My.Computer.FileSystem.CreateDirectory(System.IO.Path.GetDirectoryName(anal_filenames(0)) & "\reports")
  End If

  'Überprüfung Namen der zu erstellenden Reportdateien
  For i = 0 To amount - 1
    If System.IO.File.Exists(System.IO.Path.GetDirectoryName(anal_filenames(i)) & "\reports\" _
      & System.IO.Path.GetFileNameWithoutExtension(anal_filenames(i)) & ".rpt") Then
      System.Windows.Forms.MessageBox.Show _
        ("Reportfile ist im Zielordner bereits vorhanden! Analysenabbruch." _
        & vbCrLf & anal_filenames(i))
      DLFC_index.error_num = 1          'Analysenabbruch
      GoTo LastLine_anal
    End If
  Next

  'Analyseschleife
  For j = 0 To (amount - 1)
    refADD = "A" & System.IO.Path.GetFileNameWithoutExtension(asf_file)    'Erstellen der Referenz
    Call referenz(anal_filenames(j), refADD)    '1/1 als irrelevante übergabewerte

    camdata.Open(anal_filenames(j), 512)      'Öffnen des Spektrums
    rep_path = System.IO.Path.GetDirectoryName(anal_filenames(j)) & "\reports\" _
      & System.IO.Path.GetFileNameWithoutExtension(anal_filenames(j)) & ".rpt" 'Definition Ausgabepfad

    anal.Analyze(camdata,, asf_file,,,, rep_path)    'Analyse des Spektrums
    DUAL_DLFC.DLFC_index.PBprogress.PerformStep()    'Fortschrittsbalken aktualisieren

    camdata.Close(0)    'Schließen des Spektrums

    If DLFC_index.error_num = 1 Then GoTo LastLine_anal    'Analysenabbruch
  Next j

  DUAL_DLFC.DLFC_index.PBprogress.PerformStep()    'Fortschrittsbalken aktualisieren

Else
  System.Windows.Forms.MessageBox.Show("Analysesequenz (.asf) existiert nicht!")
  DLFC_index.error_num = 1    'Analysenabbruch
  GoTo LastLine_anal
End If

LastLine_anal:

'GUI anpassen/aktualisieren
If (amount = 1) Then
  DUAL_DLFC.DLFC_index.L_anal_stat.Text = amount & " Spektrum ausgewählt"

```

```

Else
    DUAL_DLFC.DLFC_index.L_anal_stat.Text = amount & " Spektren ausgewählt"
End If
DUAL_DLFC.DLFC_index.PBprogress.Value = 0
End Sub

Public Sub analyse_spez(anal_filenames() As String, ByVal stat As String, ByVal auto_mod As Boolean)
'Spezialanalyse für akkurate Analyse von Spektren mittels spezifizierten Parametern

Application.DoEvents()                                'Windows-Events von Form bearbeiten

DLFC_index.error_num = 0                              'Fehlervariable initialisieren
amount = anal_filenames.Count                          'Windows-Events von Form bearbeiten

Dim arr_zeilen As New ArrayList()                     'Peakenergien-Tabelle
Dim size_mid As Integer                               'Anzahl der "mittel" peaks z.B. 9 peaks
Dim size_lon As Integer                               'Anzahl der "lang" peaks z.B. 9 peaks
Dim peak_table_file As String                         'Parametertabelle-Datei
Dim date_var As Integer                              'Endzeitvariable
Dim start_time As DateTime                           'für Messung der Analysezeit pro Spektrum
'-----

peak_table_file = DUAL_DLFC.DLFC_index.TBtxt.Text     'Parametertabellenpfad übertragen

'Überprüfe Parametertabellenpfad
If (Not System.IO.File.Exists(peak_table_file)) Then
    System.Windows.Forms.MessageBox.Show("Pfad der Parameterdatei existiert nicht!")
    DLFC_index.error_num = 1                          'Analysenabbruch
    GoTo LastLine_sanal
End If

Call DUAL_DLFC.DLFC_index.Update_fct("sanal", 0)      'Update_fct aufruf

DLFC_index.error_num = check_files_dir(anal_filenames) 'Aufruf Dateüberprüfungsfunktion
If DLFC_index.error_num = 1 Then GoTo LastLine_sanal 'Analysenabbruch

arr_zeilen = get_peak_table()                         'einlesen der Parametertabelle
size_mid = arr_zeilen.IndexOf("lang:") - 3           'Definition Anzahl der zu analysierenden Energien bei "mittel"
size_lon = arr_zeilen.Count - size_mid - 5           'Definition Anzahl der zu analysierenden Energien bei "lang"

'-----Spezialanalyse-----
Dim zeilenwerte() As String                          'Zeilen für Gliederung der Parametertabelle
'-----
'-----MITTEL-----
'-----

If DUAL_DLFC.DLFC_index.RBmittel_ana.Checked Then
    Dim peaks(size_mid) As String                    'Peakenergie
    Dim KA(size_mid) As Double                       'Analyse-Peakfläche: Kanalanzfang
    Dim KE(size_mid) As Double                       'Analyse-Peakfläche: Kanalende
    Dim schwelle(size_mid) As Double                 'Analyse-Peakfläche: Residuen Grenzwert
    Dim signifikanz(size_mid) As Double              'Analyse-Peakenergie: Signifikanzschwelle
    Dim fit(size_mid) As Boolean                     'Analyse-Peakfläche: Einzelpeaks fitten?

    DUAL_DLFC.DLFC_index.PBprogress.Value = 0        'Fortschrittsbalken initialisieren
    DUAL_DLFC.DLFC_index.PBprogress.Step = Cint(100 / amount) 'Fortschrittsbalken-Schrittgröße definieren

    If (Not auto_mod) Then
        date_var = Math.Round(amount * (size_mid * 2)) '2 Sekunden für erstes Spektrum für mittlere SpezAnal pro Peak
        DLFC_index.L_auto_endzeit.Text = DateAdd(DateInterval.Second, date_var, Now).
            ToString("dd.MM.yyyy HH:mm")              'Ende der SpezAnal anzeigen
        DLFC_index.L_auto_endzeit.Visible = True     'Label anzeigen
        DLFC_index.L_auto_ende.Visible = True       'Label anzeigen
    End If

    'peak_table aufspalten in 6 arrays
    For j = 2 To size_mid + 2 '2 bis 10
        zeilenwerte = arr_zeilen(j).Split(CType(", ", Char())) 'ZEILEN TRENNEN

        If Not (zeilenwerte.Count = "6") Then
            'Formatüberprüfung
            System.Windows.Forms.MessageBox.Show("Analysenabbruch - Parametertabelle ist falsch formatiert:" & vbCrLf &
                "Das Dezimalzeichen muss ein Punkt(.), Spaltentrennzeichen ein Komma(,) sein." &
                "Die Tabelle muss 6 Spalten aufweisen!" & vbCrLf & peak_table_file)
            DLFC_index.error_num = 1                  'Analysenabbruch
            GoTo LastLine_sanal
        End If

        'Daten in arrays übertragen
        peaks(j - 2) = CType(zeilenwerte(0), String)
        KA(j - 2) = CType(zeilenwerte(1), Double)
        KE(j - 2) = CType(zeilenwerte(2), Double)
        schwelle(j - 2) = CType(zeilenwerte(3), Double)
        signifikanz(j - 2) = CType(zeilenwerte(4), Double)
        fit(j - 2) = CType(zeilenwerte(5), Boolean)
    Next

    If DLFC_index.error_num = 1 Then GoTo LastLine_sanal 'Analysenabbruch

```

```

'-----
start_time = DateTime.Now                                'Startzeit für Analysendauermessung

'Analyseschleife
For l = 0 To amount - 1                                'zu analysierende Spektren

    If ((Not auto_mod) And (l = 1)) Then
        date_var = Math.Round((amount - 1) *
            (DateDiff(DateInterval.Second, start_time, Now)))
            'Extrapolation der Analysedauer
        DLFC_index.L_auto_endzeit.Text = DateAdd(DateInterval.Second, date_var, Now).
            ToString("dd.MM.yyyy HH:mm")                'Ende der SpezAnal erneuern
    End If

    For m = 0 To size_mid                                'zu analysierende Peaks

        'Peaksuche mit 2.Ableitung
        Shell("PEAK_DIF " & """" & anal_filenames(l) & """" & " /CHANNELS=" & KA(m) & ", " & KE(m) _
            & " /SIGNIF=" & signifikanz(m), vbHide, True)

        If (fit(m)) Then
            'Integration Peakfläche - Summe / nicht.lin. LSQ FIT - mit Einzelpeakfit
            Shell("AREA_NL1 " & """" & anal_filenames(l) & """" & " /CHANNELS=" & KA(m) & ", " & KE(m) _
                & " /RES_SEARCH /RES_THRESHOLD=" & schwelle(m) & " /RES_SEPARATION=1 /FIT", vbHide, True)
        Else
            'Integration Peakfläche - Summe / nicht.lin. LSQ FIT - ohne Einzelpeakfit
            Shell("AREA_NL1 " & """" & anal_filenames(l) & """" & " /CHANNELS=" & KA(m) & ", " & KE(m) _
                & " /RES_SEARCH /RES_THRESHOLD=" & schwelle(m) & " /RES_SEPARATION=1", vbHide, True)
        End If

        'Referenzierung
        refADD = "SA" & System.IO.Path.GetFileNameWithoutExtension _
            (System.IO.Path.GetFileName(peak_table_file)) & "MIT"

        If (m = 0) Then Call referenz(anal_filenames(l), refADD)                'Funktionsaufruf
        Call report(anal_filenames(l), m, size_mid)                            'Funktionsaufruf

        If (DLFC_index.error_num = 1) Then GoTo LastLine_sanal                'Analysenabbruch
        Next

        Call DUAL_DLFC.DLFC_index.Update_fct("sanal", 1)

        'Kopie der Reportdatei von Stammordner in "reports"
        FileIO.FileSystem.CopyFile("C:\GENIE2K\REPFILES\" _
            & System.IO.Path.GetFileNameWithoutExtension(anal_filenames(l)) & ".rpt",
            System.IO.Path.GetDirectoryName(anal_filenames(l)) & "reports\" _
            & System.IO.Path.GetFileNameWithoutExtension(anal_filenames(l)) & ".rpt")
        Next

        'Überprüfung ob peaks.txt in Ordner "reports" vorhanden --> wenn ja, lösche peaks.txt
        If (System.IO.File.Exists(System.IO.Path.GetDirectoryName _
            (anal_filenames(0)) & "reports\peaks.txt")) Then
            System.IO.File.Delete(System.IO.Path.GetDirectoryName _
                (anal_filenames(0)) & "reports\peaks.txt")
        End If
        'Kopiere aktuelle peaks.txt in reports
        FileIO.FileSystem.CopyFile(peak_table_file, System.IO.Path.GetDirectoryName _
            (anal_filenames(0)) & "reports\peaks.txt")

'-----
'-----LANG-----
'-----

Elseif DUAL_DLFC.DLFC_index.RBlang_ana.Checked Then

    Dim peaks(size_lon) As String                                'Peakenergie
    Dim KA(size_lon) As Double                                'Analyse-Peakfläche: Kanalanzfang
    Dim KE(size_lon) As Double                                'Analyse-Peakfläche: Kanallende
    Dim schwelle(size_lon) As Double                        'Analyse-Peakfläche: Residuen Grenzwert
    Dim signifikanz(size_lon) As Double                    'Analyse-Peakenergie: Signifikanzschwelle
    Dim fit(size_lon) As Boolean                            'Analyse-Peakfläche: Einzelpeaks fitten?

    DUAL_DLFC.DLFC_index.PBprogress.Value = 0                'Fortschrittsbalken initialisieren
    DUAL_DLFC.DLFC_index.PBprogress.Step = CInt(100 / amount) 'Fortschrittsbalken-Schrittgröße definieren

    If (Not auto_mod) Then
        date_var = Math.Round(amount * (size_mid * 1.5))    '2 Sekunden für erstes Spektrum für mittlere SpezAnal pro Peak
        DLFC_index.L_auto_endzeit.Text = DateAdd(DateInterval.Second, date_var, Now).
            ToString("dd.MM.yyyy HH:mm")                    'Ende der SpezAnal anzeigen
        DLFC_index.L_auto_endzeit.Visible = True            'Label anzeigen
        DLFC_index.L_auto_ende.Visible = True              'Label anzeigen
    End If

    'peak_table aufspalten in 6 arrays
    For j = (arr_zeilen.IndexOf("lang:") + 1) To (arr_zeilen.Count - 1)
        zeilenwerte = arr_zeilen(j).Split(CType(", ", Char())) 'ZEILEN TRENNEN
    Next

    If Not (zeilenwerte.Count = "6") Then                    'Formatüberprüfung

```

```

System.Windows.Forms.MessageBox.Show("Analysenabbruch - Peaktabelle falsch formatiert:" & vbCrLf &
    "Das Dezimalzeichen muss ein Punkt(,), Spaltentrenner ein Komma(,) sein." &
    " Die Tabelle muss 6 Spalten haben!" & vbCrLf & peak_table_file)
DLFC_index.error_num = 1 'Analysenabbruch
GoTo LastLine_sanal
End If

'Übertragen der Daten in Arrays
peaks(j - (arr_zeilen.IndexOf("lang:") + 1)) = CType(zeilenwerte(0), String)
KA(j - (arr_zeilen.IndexOf("lang:") + 1)) = CType(zeilenwerte(1), Double)
KE(j - (arr_zeilen.IndexOf("lang:") + 1)) = CType(zeilenwerte(2), Double)
schwelle(j - (arr_zeilen.IndexOf("lang:") + 1)) = CType(zeilenwerte(3), Double)
signifikanz(j - (arr_zeilen.IndexOf("lang:") + 1)) = CType(zeilenwerte(4), Double)
fit(j - (arr_zeilen.IndexOf("lang:") + 1)) = CType(zeilenwerte(5), Boolean)
Next

If DLFC_index.error_num = 1 Then GoTo LastLine_sanal 'Analysenabbruch
'-----
start_time = DateTime.Now 'Startzeit für Analysendauermessung

'Analyseschleife
For l = 0 To anal_filenames.Count - 1 'zu analysierende Spektren

If ((Not auto_mod) And (l = 1)) Then
date_var = Math.Round((amount - 1) *
    (DateDiff(DateInterval.Second, start_time, Now))) 'Extrapolation der Analysedauer
DLFC_index.L_auto_endzeit.Text = DateAdd(DateInterval.Second, date_var, Now).
    ToString("dd.MM.yyyy HH:mm") 'Ende der SpezAnal erneuern
End If

For m = 0 To size_lon 'zu analysierende Peaks

'Peaksuche mit 2.Ableitung
Shell("PEAK_DIF " & """" & anal_filenames(l) & """" & " /CHANNELS=" & KA(m) & ", " & KE(m) _
    & " /SIGNIF=" & signifikanz(m), vbHide, True)

If (fit(m)) Then
'Integration Peakfläche - Summe / nicht.lin. LSQ FIT - mit Einzelpeakfit
Shell("AREA_NL1 " & """" & anal_filenames(l) & """" & " /CHANNELS=" & KA(m) & ", " & KE(m) _
    & " /RES_SEARCH /RES_THRESHOLD=" & schwelle(m) & " /RES_SEPARATION=1 /FIT", vbHide, True)
Else
'Integration Peakfläche - Summe / nicht.lin. LSQ FIT - ohne Einzelpeakfit
Shell("AREA_NL1 " & """" & anal_filenames(l) & """" & " /CHANNELS=" & KA(m) & ", " & KE(m) _
    & " /RES_SEARCH /RES_THRESHOLD=" & schwelle(m) & " /RES_SEPARATION=1", vbHide, True)
End If

'Referenzierung
refADD = "SA" & System.IO.Path.GetFileNameWithoutExtension _
    (System.IO.Path.GetFileName(peak_table_file)) & "LAN"

If (m = 0) Then Call referenz(anal_filenames(l), refADD) 'Funktionsaufruf
Call report(anal_filenames(l), m, size_lon) 'Funktionsaufruf

If DLFC_index.error_num = 1 Then GoTo LastLine_sanal 'Analysenabbruch
Next
Call DUAL_DLFC.DLFC_index.Update_fct("sanal", 1) 'Fortschrittsbalken - Schritt

'Kopie der Reportdatei von Stammordner in "reports"
FileIO.FileSystem.CopyFile("C:\GENIE2K\REPFILES\" _
    & System.IO.Path.GetFileNameWithoutExtension(anal_filenames(l)) & ".rpt",
    System.IO.Path.GetDirectoryName(anal_filenames(l)) & "reports\" _
    & System.IO.Path.GetFileNameWithoutExtension(anal_filenames(l)) & ".rpt")
Next

'Überprüfung ob peaks.txt in Ordner "reports" vorhanden --> wenn ja, lösche peaks.txt
If (System.IO.File.Exists(System.IO.Path.GetDirectoryName _
    (anal_filenames(0)) & "reports\peaks.txt")) Then
System.IO.File.Delete(System.IO.Path.GetDirectoryName _
    (anal_filenames(0)) & "reports\peaks.txt")
End If

'Kopiere aktuelle peaks.txt in reports
FileIO.FileSystem.CopyFile(peak_table_file, System.IO.Path.GetDirectoryName _
    (anal_filenames(0)) & "reports\peaks.txt")
End If

LastLine_sanal:
'Anpassung/Aktualisierung der GUI
If (amount = 1) Then
DUAL_DLFC.DLFC_index.L_anal_stat.Text = amount & " Spektrum ausgewählt"
Else
DUAL_DLFC.DLFC_index.L_anal_stat.Text = amount & " Spektren ausgewählt"
End If

If (Not auto_mod) Then
DLFC_index.L_auto_endzeit.Visible = False
DLFC_index.L_auto_ende.Visible = False
End If

```

```

    DUAL_DLFC.DLFC_index.PBprogress.Value = 0
End Sub

Private Sub referenz(file As String, ref_add As String)
'Referenz erstellen
Application.DoEvents()           'Windows-Events von Form bearbeiten
camdata.Open(file, 512)         'Spektrum öffnen; 512 für Read&Write

id_ref1 = ""                    'Referenz reinitialisieren
id_ref1 = camdata.Param(536936537) & camdata.Param(536936540) 'Referenz von Spektrum auslesen
id_ref1 = id_ref1 & ref_add & "/" 'Referenz modifizieren

If (id_ref1.Length <= 64) Then
    camdata.Param(536936537) = id_ref1 'Referenz speichern (Teil1)
Elseif (id_ref1.Length > 64) Then
    'Referenz speichern (Teil2)
    id_ref2 = Microsoft.VisualBasic.Right(id_ref1, id_ref1.Length - 64)
    id_ref1 = Microsoft.VisualBasic.Left(id_ref1, 64)
    camdata.Param(536936537) = id_ref1
    camdata.Param(536936540) = id_ref2
End If

camdata.Close(0)                'Spektrum schließen
End Sub

Private Sub report(file As String, m As Integer, size As Integer)
'Report erstellen
Application.DoEvents()           'Windows-Events von Form bearbeiten
Dim templ As String             'Reportvorlagendatei
templ = "sanal.tpl"             'Reportvorlage definieren

Dim sec1 As String = "Header"   'Formatvorlage - Sektion Header
Dim sec2 As String = "Tabelle"  'Formatvorlage - Sektion Tabelle
Dim sec3 As String = "Fehler"   'Formatvorlage - Sektion Fehler

camdata.Open(file, 512)         'Spektrum öffnen
If (m = 0) Then
    report_anal.ExecuteReport(camdata, templ, sec1, 4) 'Schreibe Sektion - Header einmalig
    report_anal.ExecuteReport(camdata, templ, sec2, ) 'Schreibe Sektion - Tabelle einmalig
Elseif (m > 0) Then
    report_anal.ExecuteReport(camdata, templ, sec2, ) 'Schreibe Sektion - Tabelle für alle Schleifenindizes
    If (m = size) Then report_anal.ExecuteReport(camdata, templ, sec3, ) 'Schreibe Sektion - Fehler einmalig
End If
camdata.Close(0)                'Spektrum schließen
End Sub

Private Function get_peak_table()
'EINLESEN der Parametertabelle
Application.DoEvents()           'Windows-Events von Form bearbeiten

Dim array_ges As New ArrayList() 'Initialisiere Gesamtarray für rückgabe
Dim peak_table_file As String    'Parametertabellenpfad
peak_table_file = DUAL_DLFC.DLFC_index.TBtxt.Text 'Parametertabellenpfad übertragen
Dim obj_peak_table As New StreamReader(peak_table_file) 'Initialisiere Streamreader Object für zeilenweißes einlesen
Dim zeile As String = ""

Do
    zeile = obj_peak_table.ReadLine() 'Zeile des Objekts Parametertabelle einlesen
    If Not zeile Is Nothing Then
        array_ges.Add(zeile)         'Zu Gesamtarray hinzufügen, wenn nicht leer
    End If
Loop Until zeile Is Nothing        'Bis Parametertabelle zu Ende

obj_peak_table.Close()            'Schließe Streamreader-Objekt

Return array_ges
End Function

Private Function check_files_dir(files() As String)
'Überprüfe Ordner und Dateinamen
Application.DoEvents()           'Windows-Events von Form bearbeiten

Dim amount As Integer = files.Count 'Spektranzahl definieren

'Überprüfe Order "reports"
If (Not (System.IO.Directory.Exists(System.IO.Path.GetDirectoryName(files(0)) _
    & "reports"))) Then
    My.Computer.FileSystem.CreateDirectory(System.IO.Path.GetDirectoryName(files(0)) & "reports")
End If

'Zu erstellende Dateinamen kontrollieren - wenn Dateien vorhanden --> abbrechen
For j = 0 To amount - 1
    If System.IO.File.Exists(System.IO.Path.GetDirectoryName(files(j)) & "reports\" _
    & System.IO.Path.GetFileNameWithoutExtension(files(j)) & ".rpt") Then
        System.Windows.Forms.MessageBox.Show _
        ("Reportfile ist im Zielordner bereits vorhanden! Analysenabbruch." _
        & vbCrLf & files(j))
        Return 1 'Analyseabbruch
    End If
End For

```

```

Next
Return 0
End Function
End Class

```

## TOCHTERKLASSE - INPUT\_BOX.VB

```

Public Class input_box      '{20 Zeilen}
'Benutzer-Eingabemöglichkeit
'vorrangig für Interactive Peak Fit - Dokumentation in der Referenz (ID-Key)

Dim _inp_string As String  'Eingabetext

Public Function eingabe(ByVal _spec As String) ' _spec : Name und Pfad des Spektrums
    Linput_spec.Text = _spec
    _inp_string = ""      'zurücksetzen
    TBinput.Text = ""    'zurücksetzen
    Me.ShowDialog()      'Form zur Eingabe anzeigen
    Return _inp_string   'Eingabe-String zurückgeben
End Function

Private Sub Bok_Click(sender As Object, e As EventArgs) Handles Bok.Click
    _inp_string = TBinput.Text
    Me.Close()
End Sub

End Class

```

## KLASSE - AUSWERTUNG.VB

```

Public Class auswertung      '{1486 Zeilen}
Dim i, j, k, r As Integer      'Schleifenindizes - r hinzugefügt

Public Sub auswerten(ByVal stat As String)
'Einstiegsfunktion für Klasse "auswertung".
'Alle Schritte der Auswertung werden hier durchgeführt

Application.DoEvents()      'Windows-Events von Form bearbeiten
DLFC_index.error_num = 0    'Fehlervariable initialisieren
DUAL_DLFC.DLFC_index.L_anal_stat.Visible = True      'Statusanzeige sichtbar

'Variablen definieren
Dim _filepath As String
Dim proben_data(), lang_data(), mittel_data() As String
'.....

_filepath = DUAL_DLFC.DLFC_index.TB_ordner_ausw.Text      'Pfad übertragen
DLFC_index.error_num = check_dir(_filepath)      'überprüfe und erstelle Ordner
If DLFC_index.error_num = 1 Then GoTo LastLine_ausw      'Abbruch durch Fehlerabfrage

DLFC_index.error_num = check_files(_filepath)      'überprüfe Dateien
If DLFC_index.error_num = 1 Then GoTo LastLine_ausw      'Abbruch durch Fehlerabfrage

If (DLFC_index.CBlogfile.Checked) Then      'Logfile erstellen, wenn ausgewählt
'Logfile erstellen wenn Spektren in Ordner mittel und lang vorhanden sind
If (Not System.IO.Directory.GetFiles(_filepath & "mittel\","*.cnf").Count = 0 And
Not System.IO.Directory.GetFiles(_filepath & "lang\","*.cnf").Count = 0) Then

    Call write_logfile(_filepath)      'Funktionsaufruf - Ausgeben
End If
End If

Call DLFC_index.Update_fct(stat, 0)      'Funktionsaufruf - Update

'Probanddaten einlesen: Dateien ohne erste Zeile
proben_data = get_probanddaten(_filepath & "auswertung\probanddaten.csv")      'Funktionsaufruf - Einlesen

'Nuklidlisten lang und mittel einlesen: volle Dateien
mittel_data = get_nuklide_data("C:\GENIE2K\ctfiles\nuklide_mittel.csv")      'Funktionsaufruf - Einlesen
lang_data = get_nuklide_data("C:\GENIE2K\ctfiles\nuklide_lang.csv")      'Funktionsaufruf - Einlesen

'Dimensionen übertragen
Dim proben_anz, nuk_mit_anz, nuk_lang_anz As Integer
proben_anz = proben_data.GetLength(0)
nuk_mit_anz = mittel_data.GetLength(0) - 1
nuk_lang_anz = lang_data.GetLength(0) - 1

```

```

'---Endergebnistabellen initialisieren---
Dim gesamt_erg(proben_anz, (nuk_mit_anz + nuk_lang_anz) * 2) As String      'Tabelle für auswertung.csv
Dim gesamt_messw(proben_anz, (nuk_mit_anz + nuk_lang_anz) * 2) As String    'Tabelle für messwerte.csv

If (DLFC_index.CBauto_std.CheckState = CheckState.Unchecked) Then
    DLFC_index.PBprogress.Value = 0
    DLFC_index.PBprogress.Step = CInt(100 / (nuk_mit_anz + nuk_lang_anz))
End If

'AUSWERTUNG_MITTEL durchführen
DLFC_index.error_num = evaluation_mittel(mittel_data, proben_data, _filepath, gesamt_erg, gesamt_messw)
If DLFC_index.error_num = 1 Then GoTo LastLine_ausw      'Fehlerabfrage

'AUSWERTUNG_LANG durchführen
DLFC_index.error_num = evaluation_lang(lang_data, proben_data, _filepath, gesamt_erg, gesamt_messw)
If DLFC_index.error_num = 1 Then GoTo LastLine_ausw      'Fehlerabfrage

'Gesamttabelle mit Messwerten in messwerte.csv-Datei schreiben
'Überschriftenzeile konstruieren [Mittel:      Lang:]
Dim mess_header_line As String = ",Mittel:"
For i = 1 To nuk_mit_anz * 2
    mess_header_line = mess_header_line + ", "
Next
mess_header_line = mess_header_line + "Lang:"

Call write_line(_filepath & "auswertung\messwerte.csv", mess_header_line)      'Funktionsaufruf - Überschrift, Messwerte
Call write_str_tbl(_filepath & "auswertung\messwerte.csv", gesamt_messw)      'Funktionsaufruf - Messwerttabelle

'Gesamtergebnis in auswertung.csv-Datei schreiben
'Überschriftenzeile erstellen [Mittel:      Lang:]
Dim erg_header_line As String = ",Mittel:"
For i = 1 To nuk_mit_anz * 2
    erg_header_line = erg_header_line + ", "
Next
erg_header_line = erg_header_line + "Lang:"

Call write_line(_filepath & "auswertung\auswertung.csv", erg_header_line)      'Funktionsaufruf - Überschrift, Ergebnistabelle
Call write_str_tbl(_filepath & "auswertung\auswertung.csv", gesamt_erg)      'Funktionsaufruf - Ergebnistabelle

LastLine_ausw:
DUAL_DLFC.DLFC_index.Update_fct("", 0)
DUAL_DLFC.DLFC_index.PBprogress.Value = 0
End Sub

Private Function evaluation_mittel(nuk_mittel_data(.) As String, probe_data(.) As String,
    filepath As String, ByRef ges_erg(.) As String, ByRef ges_messw(.) As String)
    'Berechnungen für Auswertung der mittlere Reportdateien durchführen

    '-----
    '-----MITTEL-----
    '-----

    'Variablen definieren
    Dim rpt1_data(.), rpt2_data(.) As Double      'Matrizen für Tabellen aus Reportdateien
    Dim rpt_acqstarttime, startzeit As DateTime      'Messstartzeit, Startzeit - Messstartzeit der 1.Probe
    Dim rpt_acqduration, cooldown As Integer      'Messlänge, Abklingzeit
    Dim std_list As New List(Of String)      'Liste der Standards und Position in Probenreihe
    Dim items(.) As String      'Alle Reportdateien
    Dim probe_nr, proben_anz, nuk_mit_anz As Integer      'Anzahlvariablen
    Dim lfc_file, unc_file As String      'Spektrienpfade
    '-----

    proben_anz = probe_data.GetLength(0)

    For j = 0 To (proben_anz - 1)
        If (Not probe_data(j, 2) = "0") Then
            std_list.Add(j & ", " & probe_data(j, 2))      'Liste der Standards mit probennr und Name erstellen
        End If
    Next

    items = IO.Directory.GetFiles(filepath & "mittel\reports", "*.rpt")      'dateinamen aller Reportdateien in ordner mittel\reports

    If (items.Count Mod 2 = 0 And proben_anz = (items.Count / 2)) Then      'anzahl der Dateien muss Gerade sein und doppelt so groß wie Probenanzahl

        'Größen definieren und Tabellen erstellen
        nuk_mit_anz = nuk_mittel_data.GetLength(0) - 1

        Dim eval_calc1_res(proben_anz - 1, nuk_mit_anz - 1) As Double      'Ergebnis der 1.Berechnung
        Dim calc_zeile(nuk_mit_anz - 1) As Double      'Zeile der Ergebnistabelle der 1.Berechnung
        Dim eval_rpt_mit(nuk_mit_anz - 1, 3) As Double      'Array für Messwerte von LFC und UNC: Peakenergie | Peakfläche
        che(LFC) | Error(UNC) | Peakfläche(UNC)
        Dim report_complete(proben_anz - 1, (nuk_mit_anz * 2) - 1) As Double      'Array für gesamten Messwertspeicher: 1 Probe pro Zeile
        Dim calc_std_res(std_list.Count + 5, nuk_mit_anz) As String      'Standardergebnistabelle - Reihen: Nuklide; Spalten: Standard-
        Rwerte + Mittelwerte/Fehler/fAbs/f%/%^2
        Dim calc_result(proben_anz, nuk_mit_anz * 2) As String      'Endergebnistabelle
        Dim obj_StreamWriter As StreamWriter      'Streamwriter Objekt für Ausgaben

```

```

Dim write_str As String                                     'Zeile für Ausgabe durch Streamwriter
Dim origin_std As Integer = 1                             'Überprüfung - Originalstandardwerte: 1 = keine daten entf.

'-----
i = 0
probe_nr = 1

'---1.SCHLEIFE: .rpt einlesen - Erste Kalkulation
Do
'-----Reportfiles einlesen-----
lfc_file = items(i)                                     'Pfad LFC-Reportdatei
unc_file = items(i + 1)                                 'Pfad UNC-Reportdatei

rpt1_data = get_rpt_table(lfc_file)                     'Reportmatrix - Spalten: Energien | Peakflächen | Fehler)
If rpt1_data(0, 0) = "111,111" Then Return 1            '111,111: Fehlercode für Reportfile - Formatfehler

rpt2_data = get_rpt_table(unc_file)                     'Reportmatrix - Spalten: Energien | Peakflächen | Fehler)
If rpt2_data(0, 0) = "111,111" Then Return 1            '111,111: Fehlercode für Reportfile - Formatfehler

rpt_acqstarttime = get_acqstarttime(lfc_file)           'Funktionsaufruf - Messstartzeit
rpt_acqduration = get_acqduration(lfc_file)             'Funktionsaufruf - Messzeit

If (i = 0) Then startzeit = rpt_acqstarttime            'Messstartzeit der 1.Probe

'-----Messwerte zusammenführen-----
For j = 1 To nuk_mit_anz

'Matrix initialisieren
eval_rpt_mit(j - 1, 0) = 0
eval_rpt_mit(j - 1, 1) = 0
eval_rpt_mit(j - 1, 2) = 0
eval_rpt_mit(j - 1, 3) = 0
'-----

For k = 0 To rpt1_data.GetLength(0) - 1                  '[0 bis rpt länge - 1]
'auszuwertende Energien, Peakflächen herausschreiben
'Für Peakenergie +- 1(nuklide_mittel.csv) = Energie der Peaks in RPT

If (CType(nuk_mittel_data(j, 2), Double) - CType(nuk_mittel_data(j, 3), Double) _
< rpt1_data(k, 0) And
CType(nuk_mittel_data(j, 2), Double) + CType(nuk_mittel_data(j, 3), Double) _
> rpt1_data(k, 0)) Then

eval_rpt_mit(j - 1, 0) = rpt1_data(k, 0)                'Peakenergie
eval_rpt_mit(j - 1, 1) = rpt1_data(k, 1)                'Peakfläche (LFC)
eval_rpt_mit(j - 1, 2) = rpt1_data(k, 2)                'Fehler(LFC)
eval_rpt_mit(j - 1, 3) = rpt1_data(k, 1)                'Peakfläche(LFC) - falls UNC Peak nicht vorhanden

End If
Next
Next

For j = 1 To nuk_mit_anz
For k = 0 To rpt2_data.GetLength(0) - 1                  '[0 bis rpt länge - 1]

'auszuwertende Fehler und unkorrigierte Peakfläche auslesen
If (CType(nuk_mittel_data(j, 2), Double) - CType(nuk_mittel_data(j, 3), Double) _
< rpt2_data(k, 0) And
CType(nuk_mittel_data(j, 2), Double) + CType(nuk_mittel_data(j, 3), Double) _
> rpt2_data(k, 0)) Then

If (Not rpt2_data(k, 2) = 0) Then
eval_rpt_mit(j - 1, 2) = rpt2_data(k, 2)                'Fehler(UNC)
End If
If (Not rpt2_data(k, 1) = 0) Then
eval_rpt_mit(j - 1, 3) = rpt2_data(k, 1)                'Peakfläche(UNC)
End If
End If
Next
Next

'Fehlerkorrektur: Fehler der LFC/UNC Korrektur muss korrigiert werden
For j = 0 To eval_rpt_mit.GetLength(0) - 1
If (eval_rpt_mit(j, 1) = 0) Then                          'Korrektur wenn Zelle nicht 0
eval_rpt_mit(j, 2) = 0
Else
eval_rpt_mit(j, 2) = Math.Sqrt(eval_rpt_mit(j, 2) + (eval_rpt_mit(j, 1) / eval_rpt_mit(j, 3) ^ 2))
End If
Next

cooldown = Convert.ToInt32(rpt_acqstarttime.Subtract(startzeit).TotalSeconds) 'Abklingzeit berechnen

'-----1.Berechnung für Zählrate pro Gramm Einwaage der Probe-----
calc_zeile = calculate_1(eval_rpt_mit, nuk_mittel_data,
probe_data, nuk_mit_anz, cooldown, rpt_acqduration, probe_nr) 'Funktionsaufruf
'-----

```

```

For j = 0 To nuk_mit_anz - 1
    eval_calc1_res(probe_nr - 1, j) = calc_zeile(j)           'zeilen zusammenfügen (proben zusammenfügen)
    report_complete(probe_nr - 1, (j * 2)) = eval_rpt_mit(j, 1) '1.Berechnung
    report_complete(probe_nr - 1, (j * 2) + 1) = eval_rpt_mit(j, 2) 'Messwerte: Peakflächen
Next
'-----
probe_nr = probe_nr + 1           'Probennummer erhöhen
i = i + 2                         'Schleifenindex erhöhen
Loop Until (i > items.Count - 1)

For i = 0 To proben_anz - 1
    ges_messw(i + 1, 0) = probe_data(i, 0)           'Messwerte in Gesamtmesswerttabelle übertragen
    'Probenbezeichnungen
    For j = 0 To nuk_mit_anz - 1
        If i = 0 Then ges_messw(0, (j * 2) + 1) = nuk_mittel_data(j + 1, 0) 'Nuklidnamen
        ges_messw(i + 1, (j * 2) + 1) = CStr(report_complete(i, j * 2)) 'Gesamtmesswerttabelle nur für Ausgabe (String)
        ges_messw(i + 1, (j * 2) + 2) = CStr(report_complete(i, (j * 2) + 1)) 'Gesamtmesswerttabelle nur für Ausgabe (String)
    Next
Next

'-----Standards berechnen-----
calc_std_res = calculate_std(eval_calc1_res, nuk_mittel_data, nuk_mit_anz, std_list)

Call write_line(filepath & "auswertung\standards_init.csv",
    "Mittel:") 'Standards Überschriftenzeile ausgeben
Call write_str_tbl(filepath & "auswertung\standards_init.csv", calc_std_res) 'unbearbeitete Standards ausgeben in standards_init.csv
Call write_line(filepath & "auswertung\standards_fin.csv",
    "Mittel:") 'Standards Überschriftenzeile ausgeben

'-----
'Standards manuell überprüfen

If (DLFC_index.CBauto_std.CheckState = CheckState.Unchecked) Then
    'Standards zur manuellen Kontrolle in neuer Form (PopUp) anzeigen

    Dim frm_std As New standards() 'Instanz der Formklasse standards erstellen
    Dim delitems(9) As Integer     'Array für gelöschte Standardwerte
    Dim dialog_stat As Integer     'Status für positive/negative Dialogantwort - wegen Shadowing
    Dim val_std(9) As String       'Originalwerte der Standards

    For j = 0 To nuk_mit_anz - 1
        frm_std.Lnuklid.Text = calc_std_res(0, j + 1) 'pro Nuklid, Standard-Form aufrufen um Werte zu prüfen
        'Nuklidname in PopUp eintragen

        For i = 1 To std_list.Count
            frm_std.Lmess.Text = "Mittel:"
            frm_std.Controls("L_std" & CStr(i)).Text = calc_std_res(i, 0) 'Namen der Std in Labels
            frm_std.Controls("TBstd" & CStr(i)).Text = calc_std_res(i, j + 1) 'Werte der Std in Textboxen
            frm_std.Controls("L_std" & CStr(i)).Visible = True 'Anzeigen
            frm_std.Controls("TBstd" & CStr(i)).Visible = True 'Anzeigen
            frm_std.Controls("B_d" & CStr(i)).Visible = True 'Anzeigen
            val_std(i) = calc_std_res(i, j + 1) 'Werte der Std in Variable übertragen
        Next

        frm_std.Lmit.Text = calc_std_res(std_list.Count + 1, j + 1) 'Mittelwert der Standards in PopUp schreiben
        frm_std.Lstabw.Text = CType(Math.Round(CType(calc_std_res(std_list.Count + 2, j + 1),
            Double) * CDBl(100), 2), String) 'STABW der Standards in PopUp schreiben

        delitems = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
        dialog_stat = frm_std.Showform(delitems, val_std)
        If dialog_stat = 0 Then '0 = OK

            For i = 0 To std_list.Count - 1
                'Daten übertragen
                If (delitems(i) = 1) Then
                    origin_std = 0
                    calc_std_res(i + 1, j + 1) = ""

                    '-----Standardwerte in standard_fin.csv schreiben-----
                    obj_StreamWriter = System.IO.File.AppendText(filepath & "auswertung\standards_fin.csv")
                    write_str = "Von Nuklid " & calc_std_res(0, j + 1) & " wurde Standard " _
                        & calc_std_res(i + 1, 0) & " entfernt!" 'Information welche Werte entfernt wurden
                    obj_StreamWriter.WriteLine(write_str) 'Ausgeben der Zeile mit StreamWriter
                    obj_StreamWriter.Close() 'Objekt schließen
                End If
            Next
        Else
            'Wenn eine Standardform mit Abbruch oder unberechtigt beendet wird
            System.Windows.Forms.MessageBox.Show("Abbruch der Auswertung. Daten nicht vollständig.")
            Return 1 'Abbruch
        End If

        Call DLFC_index.Update_fct("ausw", 1)
    Next

    'Information in standards_fin.csv, wenn keine Standardwerte entfernt wurden
    If origin_std = 0 Then
        Call write_line(filepath & "auswertung\standards_fin.csv", "")
    End If

```

```

Else
    Call write_line(filepath & "auswertung\standards_fin.csv",
        "Es wurden keine Werte der Standards entfernt.")
End If

Call recalculate_std(calc_std_res, std_list)           'Neuberechnung der Mittelwerte und Fehler
If origin_std = 0 Then
    Call write_str_tbl(filepath & "auswertung\standards_fin.csv", calc_std_res) 'bearbeitete Standards in standards_fin.csv ausgeben
End If

'-----
'Standards automatisch überprüfen

Elseif (DLFC_index.CBauto_std.CheckState = CheckState.Checked) Then
    Dim kill_value As Double           'zu löschender Wert
    Dim kill_position As Integer       'Position des zu löschenden Wertes
    Dim num_now As Integer            'aktuelle Anzahl der Werte
    Dim std_anz As Integer            'Anzahl der Standards
    Dim mittelw_pos As Integer        'Position des Mittelwertes in der Standard-Tabelle
    Dim stabw_pos As Integer         'Position der Stabw in der Standard-Tabelle

    std_anz = std_list.Count           'Anzahl der Standards
    mittelw_pos = std_anz + 1          'Position Mittelwert übertragen
    stabw_pos = std_anz + 2           'Position STABW übertragen

    For r = 1 To nuk_mit_anz
        num_now = 0
        For i = 1 To std_anz           'aktuelle Anzahl der Standard-Werte zählen
            If Not (calc_std_res(i, r) = "") Then num_now = num_now + 1
        Next

        'Abbruchkriterien, bei welcher Genauigkeit der STABW keine weiteren Standardwerte entfernt werden sollen
        If (calc_std_res(stabw_pos, r) < 0.05 Or
            calc_std_res(stabw_pos, r) < 0.0525 And num_now = 4 Or
            calc_std_res(stabw_pos, r) < 0.0575 And num_now = 3 Or
            num_now = 2) Then

            Else

                'SCHLEIFE um STABW zu verringern
                Do
                    kill_value = 0           'aktuell zu entfernender Wert
                    kill_position = 0       'Position des aktuell zu entfernenden Werts

                    For i = 1 To std_anz   'finde wert welcher am weitesten vom mittelwert entfernt
                        If Not (calc_std_res(i, r) = "") Then           'nur wenn zelle nicht leer

                            If kill_position = 0 Then                 'erster durchlauf: schreibe Differenz des Wertes und Mittelwert
                                kill_value = Math.Abs(Math.Abs(CDbI(calc_std_res(mittelw_pos, r))) -
                                    Math.Abs(CDbI(calc_std_res(i, r))))

                                kill_position = i                     'Setze Position des ersten Wertes

                            Else                                     'NICHT erster durchlauf
                                If (Math.Abs((Math.Abs(CDbI(calc_std_res(mittelw_pos, r))) -
                                    Math.Abs(CDbI(calc_std_res(i, r)))))) > kill_value) Then           'Vgl: Wenn DIFF 2.-Wert > DIFF 1.-Wert

                                    kill_value = Math.Abs(Math.Abs(CDbI(calc_std_res(mittelw_pos, r))) -
                                        Math.Abs(CDbI(calc_std_res(i, r))))           'Setze 2.Wert als zu löschenden Wert
                                    kill_position = i                               'Setze Position des 2. Wertes

                                End If
                            End If
                        End If
                    Next

                    calc_std_res(kill_position, r) = ""           'schlechtesten Wert löschen

                    Call recalculate_std(calc_std_res, std_list) 'Mittelwert/Stabw neu berechnen

                    '-----Standards in standards_fin.csv schreiben-----
                    obj_StreamWriter = System.IO.File.AppendText(filepath & "auswertung\standards_fin.csv")
                    write_str = "Von Nuklid " & calc_std_res(0, r) & " wurde Standard " _
                        & calc_std_res(kill_position, 0) & " entfernt!"           'Information welche Werte entfernt wurden
                    obj_StreamWriter.WriteLine(write_str)           'Ausgabe mittel Streamwriter
                    obj_StreamWriter.Close()                       'Streamwriter-Objekt schließen
                    '-----

                    num_now = 0
                    For i = 1 To std_anz           'Anzahl aktuell vorhandener Werte zählen
                        If Not (calc_std_res(i, r) = "") Then num_now = num_now + 1
                    Next

                    Loop Until (calc_std_res(stabw_pos, r) < 0.05 Or
                        calc_std_res(stabw_pos, r) < 0.0525 And num_now = 4 Or
                        calc_std_res(stabw_pos, r) < 0.0575 And num_now = 3 Or
                        num_now = 2)           'Schleife durchführen bis Kriterien der Genauigkeit der STABW
                End If
            End If
        Next
    Next

erreicht

```

```

        End If
    Next

    Call write_line(filepath & "auswertungstandards_fin.csv", "")
    Call write_str_tbl(filepath & "auswertungstandards_fin.csv", calc_std_res)
    'Leerzeile ausgeben
    'bearbeitete Standards in standards_fin.csv
ausgeben
End If

'-----Gesamtergebnis berechnen-----
Call calculate_erg(calc_result, report_complete, eval_calc1_res, calc_std_res, probe_data, std_list.Count)

For i = 0 To proben_anz
    For j = 0 To nuk_mit_anz * 2
        ges_erg(i, j) = calc_result(i, j)
    Next
Next
'Ergebnis in Gesamtergebnis Tabelle übertragen

Else
    System.Windows.Forms.MessageBox.Show("Unvollständige Reportfiles. " _
        & "Anzahl der Reportfiles stimmen nicht mit Probanddaten überein " _
        & "oder ist ungerade." & vbCrLf & "Abbruch der Auswertung!")
    'Fehlermeldung

    Return 1
    'Abbruch
End If

Return 0
End Function

Private Function evaluation_lang(nuk_lang_data(.) As String, probe_data(.) As String,
    filepath As String, ByRef ges_erg(.) As String, ByRef ges_messw(.) As String)
    'Berechnungen für Auswertung der langen Reportdateien durchführen
    'Detailliertere Kommentierung siehe evaluation_mittel() (Zeile 89)

    '-----LANG-----
    '-----

    Dim rpt1_data(.) As Double
    Dim rpt_acqstarttime, startzeit As DateTime
    Dim rpt_acqduration, cooldown As Integer
    Dim std_list As New List(Of String)
    Dim items() As String
    Dim probe_nr, proben_anz, nuk_lang_anz As Integer
    Dim lfc_file, unc_file As String
    '-----

    proben_anz = probe_data.GetLength(0)

    For j = 0 To (proben_anz - 1)
        If (Not probe_data(j, 2) = "0") Then
            std_list.Add(j & "," & probe_data(j, 2))
        End If
    Next
    'Liste der Standards mit probennr und Name erstellen

    items = IO.Directory.GetFiles(filepath & "lang\reports", "*.rpt")
    'in item stehen alle dateinamen von ordner mittel\reports

    If (items.Count Mod 2 = 0 And proben_anz = (items.Count / 2)) Then

        'Größen definieren und Tabellen erstellen
        nuk_lang_anz = nuk_lang_data.GetLength(0) - 1

        Dim eval_calc1_res(proben_anz - 1, nuk_lang_anz - 1) As Double
        Dim calc_zeile(nuk_lang_anz - 1) As Double
        Dim eval_rpt_lan(nuk_lang_anz - 1, 3) As Double
        Dim report_complete(proben_anz - 1, (nuk_lang_anz * 2) - 1) As Double
        Dim calc_std_res(std_list.Count + 5, nuk_lang_anz) As String
        Dim calc_result(proben_anz, nuk_lang_anz * 2) As String
        Dim obj_StreamWriter As StreamWriter
        Dim write_str As String
        Dim origin_std As Integer = 1
        'Ergebnis der 1.Berechnung
        'Array für Messwerte von LFC und UNC: Peakenergie | Peakfläche
        'Array für gesamten Messwertspeicher: 1 Probe pro Zeile
        'Standards + Namen + Mittelwerte/Fehler/fAbs/f%/%^2
        'überprüfen ob werte entfernt: 1 = keine daten entf.
    End If

    i = 0
    probe_nr = 1

    '---1.SCHLEIFE: Erste Kalkulation
    Do
        '-----Reportfiles einlesen-----

        lfc_file = items(i)
        unc_file = items(i + 1)
        rpt1_data = get_rpt_table(lfc_file)
        If rpt1_data(0, 0) = "111,111" Then Return 1
        rpt2_data = get_rpt_table(unc_file)
        If rpt2_data(0, 0) = "111,111" Then Return 1
        'Matrix mit 3 Spalten (peaks,area,error)
        '111,111: Fehlercode für Reportfile - Formatfehler
        'Matrix mit 3 Spalten (peaks,area,error)
        '111,111: Fehlercode für Reportfile - Formatfehler
    Loop
End Function

```

```

rpt_acqstarttime = get_acqstarttime(lfc_file)           'Messstartzeit
rpt_acqduration = get_acqduration(lfc_file)           'Messzeit

If (i = 0) Then startzeit = rpt_acqstarttime

'-----Messwerte zusammenführen-----
For j = 1 To nuk_lang_anz

  'Matrix initialisieren
  eval_rpt_lan(j - 1, 0) = 0
  eval_rpt_lan(j - 1, 1) = 0
  eval_rpt_lan(j - 1, 2) = 0
  eval_rpt_lan(j - 1, 3) = 0
  '-----

  For k = 0 To rpt1_data.GetLength(0) - 1
    'auszuwertende Peaks, Peakflächen herausschreiben (Für Energie +- Energie = Energie der Peaks in RPT)
    If (CType(nuk_lang_data(j, 2), Double) - CType(nuk_lang_data(j, 3), Double) _
        < rpt1_data(k, 0) And
        CType(nuk_lang_data(j, 2), Double) + CType(nuk_lang_data(j, 3), Double) _
        > rpt1_data(k, 0)) Then

      eval_rpt_lan(j - 1, 0) = rpt1_data(k, 0)           'Peakenergie
      eval_rpt_lan(j - 1, 1) = rpt1_data(k, 1)           'Peakfläche(LFC)
      eval_rpt_lan(j - 1, 2) = rpt1_data(k, 2)           'Fehler(LFC)
      eval_rpt_lan(j - 1, 3) = rpt1_data(k, 1)           'Peakfläche(LFC) - falls UNC Peak nicht vorhanden

    End If
  Next
Next
For j = 1 To nuk_lang_anz
  For k = 0 To rpt2_data.GetLength(0) - 1 '[0 bis rpt länge - 1]
    'auszuwertende Fehler herausschreiben
    If (CType(nuk_lang_data(j, 2), Double) - CType(nuk_lang_data(j, 3), Double) _
        < rpt2_data(k, 0) And
        CType(nuk_lang_data(j, 2), Double) + CType(nuk_lang_data(j, 3), Double) _
        > rpt2_data(k, 0)) Then

      If (Not rpt2_data(k, 2) = 0) Then
        eval_rpt_lan(j - 1, 2) = rpt2_data(k, 2)         'Fehler(UNC)
      End If

      If (Not rpt2_data(k, 1) = 0) Then
        eval_rpt_lan(j - 1, 3) = rpt2_data(k, 1)         'Peakfläche(UNC)
      End If
    End If
  Next
Next
'Fehlerkorrektur: Fehler der LFC/UNC Korrektur muss korrigiert werden
For j = 0 To eval_rpt_lan.GetLength(0) - 1
  If (eval_rpt_lan(j, 1) = 0) Then
    eval_rpt_lan(j, 2) = 0
  Else
    eval_rpt_lan(j, 2) = Math.Sqrt(eval_rpt_lan(j, 2) + (eval_rpt_lan(j, 1) / eval_rpt_lan(j, 3) ^ 2))
  End If
Next

cooldown = Convert.ToInt32(rpt_acqstarttime.Subtract(startzeit).TotalSeconds)

'-----1.Berechnung für Zählrate pro Gramm Einwaage der Probe-----
calc_zeile = calculate_1(eval_rpt_lan, nuk_lang_data,
    probe_data, nuk_lang_anz, cooldown, rpt_acqduration, probe_nr)
'-----

For j = 0 To nuk_lang_anz - 1
  eval_calc1_res(probe_nr - 1, j) = calc_zeile(j)           'zeilen zusammenfügen (proben zusammenfügen)
  report_complete(probe_nr - 1, (j * 2)) = eval_rpt_lan(j, 1)
  report_complete(probe_nr - 1, (j * 2) + 1) = eval_rpt_lan(j, 2)
Next

'-----
probe_nr = probe_nr + 1
i = i + 2
Loop Until (i > items.Count - 1)

For i = 0 To proben_anz - 1
  ges_messw(i + 1, 0) = probe_data(i, 0)                   'Messwerte in Gesamtmesswerttabelle übertragen
  For j = 0 To nuk_lang_anz - 1
    If i = 0 Then ges_messw(0, ges_erg.GetLength(1) - (2 * nuk_lang_anz) + (j * 2)) = nuk_lang_data(j + 1, 0)

    ges_messw(i + 1, ges_erg.GetLength(1) - (2 * nuk_lang_anz) + (j * 2)) = CStr(report_complete(i, j * 2))
    ges_messw(i + 1, ges_erg.GetLength(1) - (2 * nuk_lang_anz) + (j * 2) + 1) = CStr(report_complete(i, (j * 2) + 1))
  Next
Next

For i = 0 To proben_anz
  For j = 1 To nuk_lang_anz * 2
    'Ergebnis in Gesamtergebnis Tabelle übertragen

```

```

        ges_erg(i, j + ges_erg.GetLength(1) - 1 - (nuk_lang_anz * 2)) = calc_result(i, j)
    Next
Next
'-----Standards berechnen-----
calc_std_res = calculate_std(eval_calc1_res, nuk_lang_data, nuk_lang_anz, std_list)

Call write_line(filepath & "auswertungstandards_init.csv",
    "")
Call write_line(filepath & "auswertungstandards_init.csv",
    "Lang:")
Call write_line(filepath & "auswertungstandards_fin.csv",
    "")
Call write_line(filepath & "auswertungstandards_fin.csv",
    "Lang:")

Call write_str_tbl(filepath & "auswertungstandards_init.csv", calc_std_res)

'-----
'Standards manuell überprüfen

If (DLFC_index.CBauto_std.CheckState = CheckState.Unchecked) Then
'Standards zur manuellen Kontrolle in neuer Form (PopUp) anzeigen

Dim frm_std As New standards()
Dim delitems(9) As Integer
Dim dialog_stat As Integer
Dim val_std(9) As String

For j = 0 To nuk_lang_anz - 1
    frm_std.Lnuklid.Text = calc_std_res(0, j + 1)

    For i = 1 To std_list.Count
        frm_std.Lmess.Text = "Lang:"
        frm_std.Controls("L_std" & CStr(i)).Text = calc_std_res(i, 0)
        frm_std.Controls("TBstd" & CStr(i)).Text = calc_std_res(i, j + 1)
        frm_std.Controls("L_std" & CStr(i)).Visible = True
        frm_std.Controls("TBstd" & CStr(i)).Visible = True
        frm_std.Controls("B_d" & CStr(i)).Visible = True
        val_std(i) = calc_std_res(i, j + 1)
    Next

    frm_std.Lmit.Text = calc_std_res(std_list.Count + 1, j + 1)
    frm_std.Lstabw.Text = CType(Math.Round(CType(calc_std_res(std_list.Count + 2, j + 1),
        Double) * 100, 2), String)

    delitems = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
    dialog_stat = frm_std.Showform(delitems, val_std)
    If (dialog_stat = 0) Then

        For i = 0 To std_list.Count - 1

            If (delitems(i) = 1) Then
                origin_std = 0
                calc_std_res(i + 1, j + 1) = ""

                '-----Standards in dokumentieren-----
                obj_StreamWriter = System.IO.File.AppendText(filepath & "auswertungstandards_fin.csv")
                write_str = "Von Nuklid " & calc_std_res(0, j + 1) & " wurde Standard " _
                    & calc_std_res(i + 1, 0) & " entfernt!"
                obj_StreamWriter.WriteLine(write_str)
                obj_StreamWriter.Close()
            End If
        Next

    Else
        'Wenn eines der Standardforms unberechtigt beendet wird
        System.Windows.Forms.MessageBox.Show("Abbruch der Auswertung. Daten nicht vollständig.")
        Return 1
    End If
Next

If origin_std = 0 Then
    Call write_line(filepath & "auswertungstandards_fin.csv", "")
Else
    Call write_line(filepath & "auswertungstandards_fin.csv",
        "Es wurden keine Werte der Standards entfernt.")
End If

Call recalculate_std(calc_std_res, std_list)
If origin_std = 0 Then
    Call write_str_tbl(filepath & "auswertungstandards_fin.csv", calc_std_res)
End If

'-----
'Standards automatisch überprüfen

ElseIf (DLFC_index.CBauto_std.CheckState = CheckState.Checked) Then

```

```

Dim kill_value As Double
Dim kill_position As Integer
Dim num_now As Integer
Dim std_anz As Integer
Dim mittelw_pos As Integer
Dim stabw_pos As Integer

'zu löschender Wert
'Position des zu löschenden Wertes
'aktuelle Anzahl der Werte
'Anzahl der Standards
'Position des Mittelwertes in der Standard-Tabelle
'Position der Stabw in der Standard-Tabelle

std_anz = std_list.Count
mittelw_pos = std_anz + 1
stabw_pos = std_anz + 2

For r = 1 To nuk_lang_anz

    num_now = 0
    For i = 1 To std_anz
        If Not (calc_std_res(i, r) = "") Then num_now = num_now + 1
    Next
    'wieviele werte sind aktuell vorhanden?

    If (calc_std_res(stabw_pos, r) < 0.05 Or
        calc_std_res(stabw_pos, r) < 0.0525 And num_now = 4 Or
        calc_std_res(stabw_pos, r) < 0.0575 And num_now = 3 Or
        num_now = 2) Then

    Else
        Do
            'SCHLEIFE um Stabw zu verkleinern
            kill_value = 0
            kill_position = 0

            For i = 1 To std_anz
                If Not (calc_std_res(i, r) = "") Then
                    'finde wert welcher am weitesten vom mittelwert enf
                    'nur wenn zelle nicht leer

                    If kill_position = 0 Then
                        'erster durchlauf: schreibe Differenz des Wertes und Mittelw
                        kill_value = Math.Abs(Math.Abs(CDbI(calc_std_res(mittelw_pos, r))) -
                            Math.Abs(CDbI(calc_std_res(i, r))))

                        kill_position = i
                        'Setze Position des ersten Wertes

                    Else
                        'NICHT erster durchlauf
                        If (Math.Abs((Math.Abs(CDbI(calc_std_res(mittelw_pos, r))) _
                            - Math.Abs(CDbI(calc_std_res(i, r)))))) > kill_value) Then
                            'Vgl: Wenn DIFF 2.-Wert > DIFF 1.-Wert

                            kill_value = Math.Abs(Math.Abs(CDbI(calc_std_res(mittelw_pos, r))) _
                                - Math.Abs(CDbI(calc_std_res(i, r))))
                            'Setze 2.Wert als zu löschenden
                            kill_position = i
                            'Setze Position des 2. Wertes
                        End If
                    End If
                End If
            Next

            calc_std_res(kill_position, r) = ""
            'schlechtesten Wert löschen

            Call recalculate_std(calc_std_res, std_list)
            'Mittelwert/Stabw neu berechnen

            '-----Standards dokumentieren-----
            obj_StreamWriter = System.IO.File.AppendText(filepath & "auswertungstandards_fin.csv")
            write_str = "Von Nuklid " & calc_std_res(0, r) & " wurde Standard " _
                & calc_std_res(kill_position, 0) & " entfernt!"
            obj_StreamWriter.WriteLine(write_str)
            obj_StreamWriter.Close()
            '-----

            num_now = 0
            For i = 1 To std_anz
                If Not (calc_std_res(i, r) = "") Then num_now = num_now + 1
            Next
            'wieviele werte sind aktuell vorhanden?

            Loop Until (calc_std_res(stabw_pos, r) < 0.05 Or
                calc_std_res(stabw_pos, r) < 0.0525 And num_now = 4 Or
                calc_std_res(stabw_pos, r) < 0.0575 And num_now = 3 Or
                num_now = 2)

        End If
    Next

    Call write_line(filepath & "auswertungstandards_fin.csv", "")
    Call write_str_tbl(filepath & "auswertungstandards_fin.csv", calc_std_res)

End If

'-----Ergebnis berechnen-----
Call calculate_erg(calc_result, report_complete, eval_calc1_res, calc_std_res, probe_data, std_list.Count)

For i = 0 To proben_anz
    For j = 1 To nuk_lang_anz * 2
        ges_erg(i, j + ges_erg.GetLength(1) - 1 - (nuk_lang_anz * 2)) = calc_result(i, j)
    Next
Next

```

```

    Next
Next

Else
System.Windows.Forms.MessageBox.Show("Unvollständige Reportfiles. " _
& "Anzahl der Reportfiles stimmen nicht mit Probanddaten überein " _
& "oder ist ungerade." & vbCrLf & "Abbruch der Auswertung!")

Return 1
End If

Return 0
End Function

Private Function calculate_1(rpt(,) As Double, nuk_data(,) As String, probe(,) As String,
nuk_anz As Integer, coold As Integer, acqt As Integer, i As Integer)
'Erste Berechnung der Auswertung - Zählrate pro Gramm
'i = probennummer

Dim calc1_res(nuk_anz - 1) As Double           'Werte aller Nuklide einer Probe
Dim j As Integer                             'Schleifenindex

For j = 0 To nuk_anz - 1
If (rpt(j, 1) = 0) Then
calc1_res(j) = 0

Else
calc1_res(j) = (CType(rpt(j, 1), Double) * Math.Log(2)) _
/ (CType(nuk_data(j + 1, 1), Double) * Math.Exp(-Math.Log(2) * coold _
/ CType(nuk_data(j + 1, 1), Double)) * (1 - Math.Exp(-Math.Log(2) * acqt _
/ CType(nuk_data(j + 1, 1), Double)))) _
/ CType(probe(i - 1, 1), Double)           'Zählrate pro Gramm berechnen
End If
Next
Return calc1_res
End Function

Private Function calculate_std(calc1_res(,) As Double, nuk_data(,) As String,
nuk_anz As Integer, std_list As List(Of String))
'Berechnung R-Werte(Zählraten pro Mikrogramm) der Standards,
'den Mittelwert, die Standardabweichung sowie den Standardfehler des Mittelwerts

Dim calc_std_res(std_list.Count - 1 + 6, nuk_anz) As String 'Ergebnisarray: #STD + 5 Berechnung + Name, #Nuklide
Dim std_pos(std_list.Count - 1) As Integer 'Liste der Standardpositionen in Probenreihe
Dim std_akt As Integer 'aktueller Standard für suche in Nuklide_mittel
Dim qu_sum As Double 'Quadratsumme - dient zur Berechnung
Dim value_anz As Integer 'Anzahl der Werte

For i = 0 To std_list.Count - 1

std_pos(i) = CType(std_list(i).Split(",")(0), Integer) 'Positionen der standards in probenliste
calc_std_res(i + 1, 0) = std_list(i).Split(",")(1) 'Namen der standards in Ergebnisarray eintragen

Next

'Bezeichnungen der Zeilen unterhalb der R-Wertetabelle eintragen
calc_std_res(std_list.Count + 1, 0) = "Mittelwert"
calc_std_res(std_list.Count + 2, 0) = "Standardabweichung"
calc_std_res(std_list.Count + 3, 0) = "Fehler des Mittelwerts(FdM)"
calc_std_res(std_list.Count + 4, 0) = "FdM %"
calc_std_res(std_list.Count + 5, 0) = "FdM %^2"

'-----äußere SCHLEIFE zur Berechnung der Standards - R-Werte, Mittelwerte, Fehler
'-----Zeilen - Standards
For i = 0 To calc_std_res.GetLength(0) - 1 'Name, Standards, 5 Berechnungen

If (i <= std_list.Count - 1) Then
For k = 4 To nuk_data.GetLength(1) - 1
If (calc_std_res(i + 1, 0) = nuk_data(0, k)) Then 'Namen der Std mit denen aus Nuklid_mittel vergleichen
std_akt = k 'Position des aktuellen Standards definieren
End If
Next
End If

'-----innere SCHLEIFE zur Berechnung der Standards - R-Werte, Mittelwerte, Fehler
'-----Spalten - Nuklide
For j = 0 To nuk_anz - 1
If (i = 0) Then calc_std_res(0, j + 1) = nuk_data(j + 1, 0) 'Nuklidnamen in erste Zeile eintragen

If (i <= std_list.Count - 1) Then 'R-Werte der Standards berechnen
If (Not nuk_data(j + 1, std_akt) = "" And Not nuk_data(j + 1, std_akt) = "0" _
And Not calc1_res(std_pos(i), j) = 0) Then 'Fehlerabfrage

calc_std_res(i + 1, j + 1) = CType(calc1_res(std_pos(i), j) _
/ CType(nuk_data(j + 1, std_akt), Double), String) 'Berechnung des R-Werts

Else
calc_std_res(i + 1, j + 1) = ""

```

```

End If

Elseif (i = std_list.Count) Then
    value_anz = 0
    For k = 0 To std_list.Count - 1
        If Not (calc_std_res(k + 1, j + 1) = "") Then
            value_anz = value_anz + 1
            calc_std_res(i + 1, j + 1) = CType(CType(calc_std_res(i + 1, j + 1), Double) +
                CType(calc_std_res(k + 1, j + 1), Double), String)
        End If
    Next
    calc_std_res(i + 1, j + 1) = CType(CType(calc_std_res(i + 1, j + 1), Double) /
        CType(value_anz, Double), String)

Elseif (i = std_list.Count + 1) Then
    qu_sum = 0
    value_anz = 0

    For k = 0 To std_list.Count - 1
        If Not (calc_std_res(k + 1, j + 1) = "") Then
            value_anz = value_anz + 1
            qu_sum = CType(qu_sum +
                ((CType(calc_std_res(k + 1, j + 1), Double) -
                    CType(calc_std_res(std_list.Count + 1, j + 1),
                    Double)) ^ 2), String)
        End If
    Next

    calc_std_res(i + 1, j + 1) = CType(Math.Sqrt(qu_sum /
        CType(value_anz - 1, Double)), String)

Elseif (i = std_list.Count + 2) Then
    value_anz = 0
    For k = 0 To std_list.Count - 1
        If Not (calc_std_res(k + 1, j + 1) = "") Then
            value_anz = value_anz + 1
        End If
    Next
    calc_std_res(i + 1, j + 1) = CType(Math.Sqrt((calc_std_res(i, j + 1) ^ 2) /
        CType(value_anz, Double)), String)

    calc_std_res(i, j + 1) = CType(CType(calc_std_res(i, j + 1),
        Double) / CType(calc_std_res(std_list.Count + 1, j + 1),
        Double), String)

Elseif (i = std_list.Count + 3) Then
    calc_std_res(i + 1, j + 1) = CType(CType(calc_std_res(std_list.Count + 3, j + 1), Double) /
        CType(calc_std_res(std_list.Count + 1, j + 1), Double), String)

Elseif (i = std_list.Count + 4) Then
    calc_std_res(i + 1, j + 1) = CType((CType(calc_std_res(std_list.Count + 4, j + 1), Double) ^ 2), String)

End If
Next
Next

Return calc_std_res
End Function

Private Sub recalculate_std(ByRef std_res(.) As String, std_lst As List(Of String))
    'Neuberechnung des Mittelwerts und der Fehler der Standardtabelle
    'nach Änderung der R-Werte durch den Benutzer

    Dim std_pos(std_lst.Count - 1) As Integer
    Dim qu_sum As Double
    Dim value_anz As Integer

    'neu zu berechnende Zellen leeren
    For i = std_lst.Count + 1 To std_res.GetLength(0) - 1
        For j = 1 To std_res.GetLength(1) - 1
            std_res(i, j) = "0"
        Next
    Next

    '-----äußere SCHLEIFE zur Berechnung der Standards - R-Werte, Mittelwerte, Fehler
    'Zeilen - Standards
    For i = 0 To std_res.GetLength(0) - 1
        '-----innere SCHLEIFE zur Berechnung der Standards - Mittelwerte, Fehler
        'Spalten - Nuklide
        For j = 0 To std_res.GetLength(1) - 2
            If (i = std_lst.Count) Then
                value_anz = 0

                For k = 0 To std_lst.Count - 1
                    If Not (std_res(k + 1, j + 1) = "") Then

```

```

        value_anz = value_anz + 1
        std_res(i + 1, j + 1) = CType(std_res(i + 1, j + 1), Double) +
        CType(std_res(k + 1, j + 1), Double) 'Berechnung
    End If
Next

std_res(i + 1, j + 1) = CType(CType(std_res(i + 1, j + 1), Double) /
    CType(value_anz, Double), String) 'Division der Summe durch Anzahl der Werte

Elseif (i = std_lst.Count + 1) Then 'Standardabweichung der R-Werte vom Mittelwert berechnen
    qu_sum = 0
    value_anz = 0

    For k = 0 To std_lst.Count - 1
        If Not (std_res(k + 1, j + 1) = "") Then 'Nur wenn Zelle nicht leer
            value_anz = value_anz + 1
            qu_sum = CType(qu_sum + ((CType(std_res(k + 1, j + 1), Double) -
                CType(std_res(std_lst.Count + 1, j + 1),
                Double)) ^ 2), String) 'Quadratsumme der Abweichungen der Werte vom Mittelwert
        End If
    Next

    std_res(i + 1, j + 1) = CType(Math.Sqrt(qu_sum /
        CType(value_anz - 1, Double)), String) 'Berechnung der absoluten standardabweichung
    '->vorerst ohne division durch den mittelwert, da stabw im nächsten schritt nochmal verwendet wird

Elseif (i = std_lst.Count + 2) Then 'Standardfehler des mittelwerts (absolut) berechnen
    value_anz = 0
    For k = 0 To std_lst.Count - 1
        If Not (std_res(k + 1, j + 1) = "") Then 'Nur wenn Zelle nicht leer
            value_anz = value_anz + 1
        End If
    Next
    std_res(i + 1, j + 1) = CType(Math.Sqrt((std_res(i, j + 1) ^ 2) /
        CType(value_anz, Double)), String) 'Standardfehler des mittelwerts (absolut) berechnen

    std_res(i, j + 1) = CType(CType(std_res(i, j + 1), Double) /
        CType(std_res(std_lst.Count + 1, j + 1), Double), String) 'STABW in Prozent umrechnen

Elseif (i = std_lst.Count + 3) Then 'Fehler% berechnen
    std_res(i + 1, j + 1) = CType(CType(std_res(std_lst.Count + 3, j + 1), Double) /
        CType(std_res(std_lst.Count + 1, j + 1), Double), String)

Elseif (i = std_lst.Count + 4) Then '(Fehler%)^2 berechnen
    std_res(i + 1, j + 1) = CType((CType(std_res(std_lst.Count + 4, j + 1), Double) ^ 2), String)
End If
Next
Next
End Sub

Private Sub calculate_erg(ByRef result(.) As String, ByVal report(.) As Double, ByVal calc1(.) As Double,
    ByVal calc_std(.) As String, proben(.) As String, std_anz As Integer)
    'Berechnung des Endergebnisses

    Dim i, j, k As Integer 'Schleifenindizes

    For k = 1 To calc_std.GetLength(1) - 1
        result(0, (k * 2) - 1) = calc_std(0, k) 'Nuklidnamen in result eintragen
    Next
    For k = 0 To proben.GetLength(0) - 1
        result(k + 1, 0) = proben(k, 0) 'Probenamen in result eintragen
    Next

    '-----äußere SCHLEIFE: Zeilen - Proben
    For i = 0 To result.GetLength(0) - 2

        '-----innere SCHLEIFE: Spalten - Nuklide
        For j = 1 To calc_std.GetLength(1) - 1
            result(i + 1, (j * 2) - 1) = CType(calc1(i, j - 1) /
                CType(calc_std(std_anz + 1, j), Double), String) 'µg/g Berechnen

            If (Not report(i, (j * 2) - 1) = 0) Then 'wenn Integrationsfehler nicht 0
                result(i + 1, (j * 2)) = CType(Math.Sqrt(CType(
                    (calc_std(std_anz + 5, j), Double) +
                    (report(i, (j * 2) - 1) / CDb(100)) ^ 2), String) 'Fehler berechnen

                result(i + 1, (j * 2)) = CType(CType(result(i + 1, (j * 2) -
                    1), Double) * CType(result(i + 1, (j * 2)), Double), String) 'Relativen Fehler in absoluten Fehler umrechnen
            Else
                result(i + 1, (j * 2)) = CStr(0) 'Fehler = 0, wenn Integrationsfehler = 0
            End If
        Next
    Next
Next
End Sub

Private Sub write_str_tbl(fp As String, str_tbl(.) As String)

```

```

'Schreibe gesamte Stringtabelle in Ausgabedatei

Dim obj_StreamWriter As StreamWriter           'Streamwriter-Objekt initialisieren
Dim write_str As String                       'Ausgabezeile initialisieren
Dim i, j As Integer                           'Schleifenindizes

obj_StreamWriter = System.IO.File.AppendText(fp)

For i = 0 To str_tbl.GetLength(0) - 1

    write_str = Replace(str_tbl(i, 0), ",", ".") 'erste Zelle der akt. Zeile inkl. Austausch des Dezimalzeichen

    For j = 1 To str_tbl.GetLength(1) - 1
        write_str = write_str & "," & Replace(str_tbl(i, j), ",", ".") 'erweitere Ausgabezeile mit Zellen der gesamten Zeile inkl. Austausch
des Dez. Zeichens
    Next

    obj_StreamWriter.WriteLine(write_str)      'Schreibe Ausgabezeile
Next
obj_StreamWriter.Close()                     'SchlieÙe StreamWriter-Objekt
End Sub

Private Sub write_logfile(_directory As String)
'Erstelle Dokumentationsdatei als Überblick aller Referenzen der ausgewerteten Spektren

Application.DoEvents()                       'Windows-Events von Form bearbeiten
Dim camdata As New CanberraDataAccessLib.DataAccess 'neue Instanz der Canberra-DataAccess-Klasse

DLFC_index.error_num = 0                     'Fehlervariable initialisieren

'Variablen initialisieren
Dim spec_files() As String                   'Pfade der Spektren
Dim p As Integer                             'Schleifenindizes
Dim obj_StreamWriter As StreamWriter         'StreamWriter-Objekt - f. Konstruktion der Ausgabedatei
Dim write_str As String                       'zu schreibende Zeile
Dim id_ref1 As String                         'Referenz

obj_StreamWriter = System.IO.File.AppendText(_directory & "auswertung\logfile.csv") 'StreamWriter-Objekt: Zieldatei

'Header der logfile.csv Datei
write_str = ""                               'Ausgabezeile initialisieren
write_str = "Spektrenreferenzen bei Auswertung am: " _
    & DateTime.Now().ToString("dd.MM.yyyy HH:mm") 'Header konstruieren
obj_StreamWriter.WriteLine(write_str)         'Ausgabe in Datei

write_str = ""                               'Ausgabezeile leeren
obj_StreamWriter.WriteLine(write_str)         'Ausgabe in Datei

write_str = "Spektrename,Referenz"          'Überschriftenzeile konstruieren
obj_StreamWriter.WriteLine(write_str)        'Ausgabe in Datei

'MITTLERE Spektren auslesen und in Referenz in Logfile übertragen
_directory = DUAL_DLFC.DLFC_index.TB_ordner_ausw.Text
spec_files = IO.Directory.GetFiles(_directory & "mittel", "*.cnf")

DUAL_DLFC.DLFC_index.PBprogress.Value = 0    'Fortschrittsbalken initialisieren
DUAL_DLFC.DLFC_index.PBprogress.Step = CInt(100 / 8) 'Fortschrittsbalken-Schrittgröße definieren

For p = 0 To spec_files.Count - 1
    If DLFC_index.error_num = 1 Then GoTo Lastline_log 'Error: Abbruch
    camdata.Open(spec_files(p), 512)             'Spektrum öffnen; 512 für Read&Write

    id_ref1 = ""                                 'Referenz reinitialisieren
    id_ref1 = camdata.Param(536936537) & camdata.Param(536936540) 'Referenz von Spektrum auslesen
    camdata.Close(0)                             'Spektrum schließen

    write_str = ""                               'Ausgabezeile leeren
    write_str = System.IO.Path.GetFileName(spec_files(p)) & "," 'Dateiname eintragen und "," als Trennzeichen
    write_str = write_str & Replace(id_ref1, ",", ".") 'Referenz hinzufügen (ungewollte Trennzeichen vermeiden)
    obj_StreamWriter.WriteLine(write_str)         'Ausgabe in Datei
    If (p = CInt(spec_files.Count / 4) Or
        p = CInt(spec_files.Count / 2) Or
        p = CInt(spec_files.Count * 3 / 4) Or
        p = spec_files.Count) Then
        Call DLFC_index.Update_fct("logf", 1) 'Fortschrittsbalken - Schritt
    End If
Next

write_str = ""                               'Ausgabezeile leeren
obj_StreamWriter.WriteLine(write_str)         'Ausgabe in Datei

'LANGE Spektren auslesen und in Referenz in Logfile übertragen
write_str = "Lang:"
obj_StreamWriter.WriteLine(write_str)         'Ausgabe in Datei

_directory = DUAL_DLFC.DLFC_index.TB_ordner_ausw.Text
spec_files = IO.Directory.GetFiles(_directory & "lang", "*.cnf")

```

```

For p = 0 To spec_files.Count - 1
  If DLFC_index.error_num = 1 Then GoTo Lastline_log
  camdata.Open(spec_files(p), 512)
  'Error: Abbruch
  'Spektrum öffnen; 512 für Read&Write

  id_ref1 = ""
  id_ref1 = camdata.Param(536936537) & camdata.Param(536936540)
  camdata.Close(0)
  'Referenz reinitialisieren
  'Referenz von Spektrum auslesen
  'Spektrum schließen

  write_str = System.IO.Path.GetFileName(spec_files(p)) & ","
  write_str = write_str & Replace(id_ref1, ",", ".")
  obj_StreamWriter.WriteLine(write_str)
  'Dateiname eintragen und "," als Trennzeichen
  'Referenz hinzufügen (ungewollte Trennzeichen vermeiden)
  'Ausgabe in Datei
  If (p = CInt(spec_files.Count / 4) Or
      p = CInt(spec_files.Count / 2) Or
      p = CInt(spec_files.Count * 3 / 4) Or
      p = spec_files.Count) Then
    Call DLFC_index.Update_fct("log", 1)
    'Fortschrittsbalken - Schritt
  End If
Next

obj_StreamWriter.Close()

Lastline_log:

End Sub

Private Sub write_line(fp As String, line As String)
  'Schreibe einzelne Zeile in Zieldatei

  Dim obj_StreamWriter As StreamWriter
  'Initialisiere Streamwriter

  obj_StreamWriter = System.IO.File.AppendText(fp)
  obj_StreamWriter.WriteLine(line)
  obj_StreamWriter.Close()
  'Definiere Zieldatei
  'Schreibe Ausgabezeile
  'SchlieÙe StreamWriter-Objekt

End Sub

Private Function check_dir(filepath As String)
  'Überprüfung der Ordnerstruktur

  'Abbruch wenn Auswertungsordner nicht existiert
  If (Not (System.IO.Directory.Exists(filepath & "auswertung"))) Then
    System.Windows.Forms.MessageBox.Show("Auszuwertender Ordner existiert nicht!" & vbCrLf _
      & "Abbruch der Auswertung.")
    Return 1
    'Fehlerrückgabe
  Else

    'Abbruch wenn Ordner "mittel" und "reports" nicht vorhanden
    If (Not (System.IO.Directory.Exists(filepath & "mittel"))) Then
      System.Windows.Forms.MessageBox.Show("Reportfiles der mittleren Messreihe nicht verfügbar!" & vbCrLf _
        & "Abbruch der Auswertung.")
      Return 1
      'Fehlerrückgabe
    Else
      If (Not (System.IO.Directory.Exists(filepath & "mittel\reports"))) Then
        System.Windows.Forms.MessageBox.Show("Reportfiles der mittleren Messreihe nicht verfügbar!" & vbCrLf _
          & "Abbruch der Auswertung.")
        Return 1
        'Fehlerrückgabe
      End If
    End If

    'Abbruch wenn Ordner "lang" und "reports" nicht vorhanden
    If (Not (System.IO.Directory.Exists(filepath & "lang"))) Then
      System.Windows.Forms.MessageBox.Show("Reportfiles der langen Messreihe nicht verfügbar!" & vbCrLf _
        & "Abbruch der Auswertung.")
      Return 1
    Else
      If (Not (System.IO.Directory.Exists(filepath & "lang\reports"))) Then
        System.Windows.Forms.MessageBox.Show("Reportfiles der langen Messreihe nicht verfügbar!" & vbCrLf _
          & "Abbruch der Auswertung.")
        Return 1
      End If
    End If

  End If

  Return 0
  'positive Rückgabe
End Function

Private Function check_files(filepath As String)
  'Überprüfung der notwendigen Analysedateien

  Dim items() As String
  'Array aller Pfade der Reportdateien

  'Abbruch wenn benötigte Dateien zur Auswertung nicht vorhanden sind
  items = IO.Directory.GetFiles(filepath & "mittel\reports")
  'MITTEL - Reportdateien speichern

```

```

If (items.Count = 0) Then
    System.Windows.Forms.MessageBox.Show("Reportfiles der mittleren Messreihe nicht verfügbar!" & vbCrLf _
        & "Abbruch der Auswertung.")
    Return 1 'Fehlerrückgabe wenn keine mittleren Reports vorhanden
End If

items = IO.Directory.GetFiles(filepath & "lang\reports") 'LANG - Reportdateien speichern
If (items.Count = 0) Then
    System.Windows.Forms.MessageBox.Show("Reportfiles der langen Messreihe nicht verfügbar!" & vbCrLf _
        & "Abbruch der Auswertung.")
    Return 1 'Fehlerrückgabe wenn keine langen Reports vorhanden
End If

If (Not System.IO.File.Exists(filepath & "auswertung\probendaten.csv")) Then
    System.Windows.Forms.MessageBox.Show _
        ("Fehlende Probendaten! Abbruch der Auswertung." _
        & vbCrLf & filepath & "auswertung\probendaten.csv")

    Return 1 'Fehlerrückgabe wenn probendaten.csv nicht vorhanden

Elseif (Not System.IO.File.Exists("C:\GENIE2K\ctfiles\nuklide_lang.csv")) Then
    System.Windows.Forms.MessageBox.Show _
        ("Fehlende Nuklidliste! Abbruch der Auswertung." _
        & vbCrLf & "C:\GENIE2K\ctfiles\nuklide_lang.csv")

    Return 1 'Fehlerrückgabe wenn nuklide_lang.csv nicht vorhanden

Elseif (Not System.IO.File.Exists("C:\GENIE2K\ctfiles\nuklide_mittel.csv")) Then
    System.Windows.Forms.MessageBox.Show _
        ("Fehlende Nuklidliste! Abbruch der Auswertung." _
        & vbCrLf & "C:\GENIE2K\ctfiles\nuklide_mittel.csv")

    Return 1 'Fehlerrückgabe wenn nuklide_mittel.csv nicht vorhanden
End If

'Abbruch wenn Ausgabedateien bereits vorhanden
If System.IO.File.Exists(filepath & "auswertung\auswertung.csv") Then
    System.Windows.Forms.MessageBox.Show _
        ("Ausgabedatei ist im Zielordner bereits vorhanden! Abbruch der Auswertung." _
        & vbCrLf & filepath & "auswertung\auswertung.csv")

    Return 1 'Fehlerrückgabe wenn auswertung.csv bereits vorhanden

Elseif System.IO.File.Exists(filepath & "auswertung\standards_init.csv") Then
    System.Windows.Forms.MessageBox.Show _
        ("Ausgabedatei ist im Zielordner bereits vorhanden! Abbruch der Auswertung." _
        & vbCrLf & filepath & "auswertung\standards_init.csv")

    Return 1 'Fehlerrückgabe wenn standards_init.csv bereits vorhanden

Elseif System.IO.File.Exists(filepath & "auswertung\standards_fin.csv") Then
    System.Windows.Forms.MessageBox.Show _
        ("Ausgabedatei ist im Zielordner bereits vorhanden! Abbruch der Auswertung." _
        & vbCrLf & filepath & "auswertung\standards_fin.csv")

    Return 1 'Fehlerrückgabe wenn standards_fin.csv bereits vorhanden

Elseif System.IO.File.Exists(filepath & "auswertung\messwerte.csv") Then
    System.Windows.Forms.MessageBox.Show _
        ("Ausgabedatei ist im Zielordner bereits vorhanden! Abbruch der Auswertung." _
        & vbCrLf & filepath & "auswertung\messwerte.csv")

    Return 1 'Fehlerrückgabe wenn messwerte.csv bereits vorhanden

Elseif System.IO.File.Exists(filepath & "auswertung\logfile.csv") Then
    System.Windows.Forms.MessageBox.Show _
        ("Ausgabedatei ist im Zielordner bereits vorhanden! Abbruch der Auswertung." _
        & vbCrLf & filepath & "auswertung\logfile.csv")

    Return 1 'Fehlerrückgabe wenn logfile.csv bereits vorhanden
End If

Return 0 'positiver Rückgabewert
End Function

Public Function get_probendaten(file As String)
    'Datei probendaten.csv einlesen mit StreamReader-Klasse

    Application.DoEvents() 'Windows-Events von Form bearbeiten
    Dim data_file As New StreamReader(file) 'StreamReader-Klasse
    Dim zeile As String = "" 'Zeile initialisieren
    Dim arr_zeilen As New ArrayList() 'Array aller eingelesenen Zeilen
    Dim zeilenwerte() As String 'einzelne Werte einer Zeile
    Dim j As Integer 'Schleifenindex
    Dim data_size As Integer 'Zeilenanzahl

    '---Zeilenweises EINLESEN---

```

```

Do While Not (zeile Is Nothing)
    zeile = data_file.ReadLine()
    If Not (zeile Is Nothing) Then
        arr_zeilen.Add(zeile)          'Erstellen einer Liste aller Zeilen
    End If
Loop
data_file.Close()

data_size = arr_zeilen.Count          'Zeilenanzahl definieren
Dim data(data_size - 2, 2) As String  '-2 weil index von 0 - datasize; erste zeile in probedata = namen

'Tabelle einlesen und in arrays aufteilen
For j = 0 To data_size - 2

    zeilenwerte = arr_zeilen(j + 1).Split(CType(", ", Char()))  'ZEILEN TRENNEN
    data(j, 0) = Replace(zeilenwerte(0), " ", ",")              'Probenbezeichnung
    data(j, 1) = Replace(zeilenwerte(1), " ", ",")              'Masse der Probeneinwaage
    data(j, 2) = Replace(zeilenwerte(2), " ", ",")              'Standardbezeichnung bzw. 0=kein Standard
Next

Return (data)                    'Rückgabe der probendaten Tabelle
End Function

Public Function get_nuklide_data(file As String)
    'Einlesen der Dateien nuklide_mittel.csv und nuklide_lang.csv mittels Streamreader-Klasse

    Application.DoEvents()        'Windows-Events von Form bearbeiten
    Dim data_file As New StreamReader(file)  'Streamreader-Klasse
    Dim zeile As String = ""        'Zeile initialisieren
    Dim arr_zeilen As New ArrayList()  'Array aller eingelesenen Zeilen
    Dim zeilenwerte() As String      'einzelne Werte einer Zeile
    Dim j As Integer                 'Schleifenindex
    Dim data_rows As Integer         'Zeilenanzahl
    Dim data_columns As Integer      'Spaltenanzahl

    '---Zeilenweises EINLESEN---
    Do While Not (zeile Is Nothing)
        zeile = data_file.ReadLine()
        If Not (zeile Is Nothing) Then
            arr_zeilen.Add(zeile)    'Erstellen einer Liste aller Zeilen
        End If
    Loop
    data_file.Close()              'Datei schließen

    data_rows = arr_zeilen.Count    'Zeilenzahl definieren
    zeilenwerte = arr_zeilen(0).split(CType(", ", Char()))  'Zeilen trennen
    data_columns = zeilenwerte.Count  'Spaltenzahl definieren

    Dim data(data_rows - 1, data_columns - 1) As String  'index von 0 - datasize

    'Tabelle einlesen und in arrays aufteilen
    For j = 0 To data_rows - 1
        zeilenwerte = arr_zeilen(j).Split(CType(", ", Char()))  'ZEILEN TRENNEN und übergeben
        For i = 0 To data_columns - 1
            data(j, i) = Replace(zeilenwerte(i), ".", ",")  'Dezimalzeichen ersetzen
        Next
    Next

    Return (data)                    'Tabelle zurückgeben
End Function

Public Function get_rpt_table(file As String)
    'Einlesen der Reportdateien

    Application.DoEvents()        'Windows-Events von Form bearbeiten
    Dim data_file As New StreamReader(file)  'Streamreader- Objekt initialisieren
    Dim zeile As String = ""        'Zeile initialisieren
    Dim arr_zeilen As New ArrayList()  'Zeilenliste
    Dim data_size As Integer        'Dateigröße
    Dim table_size As Integer       'Wertetabellengröße
    Dim zeilenwerte() As String     'nicht initialisiertes array - für Formatüberprüfung
    Dim j As Integer                 'Schleifenindex

    '---zeilenweises EINLESEN---
    Do
        zeile = data_file.ReadLine()
        If Not zeile Is Nothing Then
            arr_zeilen.Add(zeile)
        End If
    Loop Until Left(zeile, 13) = "Errors quoted"  'Einlesen bis Terminator

    data_file.Close()              'Datei schließen
    data_size = arr_zeilen.Count    'Zeilenanzahl der Datei übergeben
    table_size = data_size - 22     'Messwerttabellengröße = Gesamtdateigröße - Headerzeilen(22)

    Dim erg_return(table_size - 1, 2) As Double  'peak | area | error

    'Tabelle einlesen und in arrays aufteilen

```

```

For j = 20 To data_size - 3                                'ab 20 wegen HEADER von Report

arr_zeilen(j) = Replace(arr_zeilen(j), ",", ",")          'WICHTIG wenn region einstellung bei pc österreich(deutsch)
Dim fst As Char = arr_zeilen(j)(0)

'Kürzen der Zeile
Do
arr_zeilen(j) = Right(arr_zeilen(j),
(arr_zeilen(j).ToString.Length - 1))                    'zeile kürzen bis zahl 1-10 an erster stelle(multipllett mit max. 10 peaks möglich)
fst = arr_zeilen(j)(0)
Loop Until (fst = "F" Or fst = "M" Or fst = "m" Or fst = "1" Or fst = "2" Or fst = "3" Or
fst = "4" Or fst = "5" Or fst = "6" Or fst = "7" Or fst = "8" Or fst = "9" Or
Left(arr_zeilen(j), 13) = "or more peaks")              'kürzen bis Abbruchkriterium

If (Left(arr_zeilen(j), 13) = "or more peaks") Then
arr_zeilen(j) = "error"                                  'Ausnahme für Peak-Drop Fehler während Analyse

Elseif (fst = "F" Or fst = "M" Or fst = "m") Then
arr_zeilen(j) = Right(arr_zeilen(j),
(arr_zeilen(j).ToString.Length - 1))                    'fst = F/M/m abschneiden
End If

zeilenwerte = arr_zeilen(j).Split(CType("", Char()),
StringSplitOptions.RemoveEmptyEntries)                    'ZEILEN TRENNEN

If (arr_zeilen(j) = "error") Then                        'Zeile 0 setzen bei Fehler
erg_return(j - 20, 0) = 0
erg_return(j - 20, 1) = 0
erg_return(j - 20, 2) = 0

Else
If Not (zeilenwerte.Count = "7") Then                   'Formatüberprüfung durch Kontrolle der Spaltenanzahl
System.Windows.Forms.MessageBox.Show("Abbruch der Auswertung - Formatfehler in Reportfile:" & vbCrLf &
"Bsp.: M/m/F werden abgeschnitten. " & vbCrLf &
"Restliche Zeile muss 7 Spalten, mit Leerzeichen getrennt, enthalten!" &
vbCrLf & file)
DLFC_index.error_num = 1
erg_return(0, 0) = "111,111"                            '111,111: Fehlercode für Reportfile - Formatfehler

Return erg_return                                       'Rückgabe mit Fehlercode
End If

erg_return(j - 20, 0) = CType(zeilenwerte(1), Double)  'Peakenergie
erg_return(j - 20, 1) = CType(zeilenwerte(2), Double)  'Peakfläche

If (zeilenwerte(6) = "*****") Then
zeilenwerte(6) = 0                                       'fehlerhaften Fehler 0 setzen
End If

erg_return(j - 20, 2) = CType(zeilenwerte(6), Double)  'Fehler
End If
Next

Return erg_return                                       'erfolgreiche Rückgabe
End Function

Public Function get_acqstarttime(file As String)
'Messstartzeit von Reportfile auslesen

Application.DoEvents()                                  'Windows-Events von Form bearbeiten
Dim data_file As New StreamReader(file)                  'StreamReader-Objekt initialisieren
Dim zeile As String = ""                                'Zeile initialisieren
Dim arr_zeilen As New ArrayList()                       'Zeilenliste
Dim startzeit As DateTime                               'Messstartzeit

Dim datum_split(2) As String                            'Datumsteile
Dim zeit As DateTime                                    'Zeit des Messstarts
Dim midnight As DateTime = "00:00:00"                  'Mitternacht
Dim zeilenwerte() As String                             'array für Werte einer Zeile

'---EINLESEN---
Do
zeile = data_file.ReadLine()
If Not zeile Is Nothing Then
arr_zeilen.Add(zeile)                                  'Zeilenliste erstellen
End If
Loop Until arr_zeilen.Count = 12

data_file.Close()                                       'Datei schließen

'Startzeit einlesen
zeilenwerte = arr_zeilen(9).split(CType("", Char()),
StringSplitOptions.RemoveEmptyEntries)                    'Werte der Zeile 9 trennen
datum_split = zeilenwerte(3).Split(CType(".", Char()))  'vierter Wert von Zeile 9 durch "." trennen
datum_split(2) = "20" & datum_split(2)                 'Datumsformat vervollständigen "2018"
Dim datum As New Date(CType(datum_split(2), Integer), CType_
(datum_split(1), Integer), CType(datum_split(0), Integer), 0, 0, 0) 'Datum aus Einzelteilen zusammensetzen

```

```

    zeit = CType(zeilenwerte(4), DateTime)           'Zeit aus Messstart übergeben
    startzeit = datum.Add(zeit - midnight)          'Messstartzeit /-datum zusammensetzen
    Return (startzeit)                              'Messstartzeit übergeben
End Function

Public Function get_acqduration(file As String)
    'Messdauer aus Reportfile auslesen

    Application.DoEvents()                         'Windows-Events von Form bearbeiten
    Dim data_file As New StreamReader(file)         'StreamReader-Objekt initialisieren
    Dim zeile As String = ""                       'Zeile initialisieren
    Dim arr_zeilen As New ArrayList()             'Zeilenliste
    Dim messzeit As Integer                       'Messdauer
    Dim zeilenwerte() As String                  'Array für Werte einer Zeile

    '---EINLESEN---
    Do
        zeile = data_file.ReadLine()
        If Not zeile Is Nothing Then
            arr_zeilen.Add(zeile)                 'Zeilenliste erstellen
        End If
    Loop Until arr_zeilen.Count = 12

    data_file.Close()                             'Datei schließen

    'Messzeit einlesen
    zeilenwerte = arr_zeilen(10).split(CType("", Char()), 'Zeilenwerte trennen
        StringSplitOptions.RemoveEmptyEntries)
    messzeit = zeilenwerte(2)                    'Messdauer übergeben
    Return (messzeit)                            'Rückgabewert
End Function

End Class

```

## TOCHTERKLASSE - STANDARDS.VB

```

Public Class standards '{177 Zeilen}
    'Form zur Anzeige der Radionuklide mit allen vorhandenen Standards bzw. deren Zählraten
    'Möglichkeit zur Entfernung von Werten

    Dim deleted_items = New Integer() {0, 0, 0, 0, 0, 0, 0, 0, 0, 0} 'Übergabe der zu entfernenden Werte
    Dim delete_stack As New List(Of Integer) 'Stapel für Entfernen - Rückgängig
    Dim all_val(9) As String 'Alle Standard-Anfangswerte

    Public Function Showform(ByRef data() As Integer, ByVal _val_std() As String)
        'Zeige Form an (Individuelle ShowDialog Funktion)

        all_val = _val_std
        deleted_items = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0} 'Initialisieren
        If Me.ShowDialog() = Windows.Forms.DialogResult.OK Then 'Form wird als Dialog angezeigt -> DialogResult=OK muss möglich sein
            data = deleted_items 'Gelöschte Werte übergeben - ByRef
            Return 0
        Else
            Return 1 'Abbruch
        End If
    End Function

    Private Sub Bundo_Click(sender As Object, e As EventArgs) Handles Bundo.Click
        Me.Controls("TBstd" & delete_stack.Last).Text = all_val(delete_stack.Last) 'Setze Wert wieder in Textbox
        deleted_items(delete_stack.Last - 1) = 0 'Setze zu löschenden Wert auf 0(False)
        delete_stack.Remove(delete_stack.Last) 'Entferne vom Löschstapel betreffenden letzten Wert
        Call recalc() 'Neuberechnung von Mittelwert und Stabw
        If (delete_stack.Count = 0) Then Bundo.Enabled = False 'Deaktiviere [Rückgängig], wenn kein Wert entfernt wurde
    End Sub

    Public Sub Bsave_Click(sender As Object, e As EventArgs) Handles Bsave.Click
        Me.DialogResult = System.Windows.Forms.DialogResult.OK 'für positives Dialog.Result
        Me.Close()
    End Sub

    Private Sub Bcancel_Click(sender As Object, e As EventArgs) Handles Babbruch.Click
        Me.DialogResult = System.Windows.Forms.DialogResult.Cancel 'für negatives Dialog.Result
        Me.Close()
    End Sub

    Private Sub recalc()
        'Neuberechnung des Mittelwert's und der Standardabweichung

        Dim value_anz, i As Integer
        Dim res_mit As Double
        Dim res_stabw As Double
        Dim count As Integer

```

```

Dim qu_sum As Double
'-----

count = 0
value_anz = 0
For i = 1 To 10
    If (Controls("TBstd" & CStr(i)).Visible = True) Then
        count = count + 1          'Anzahl der Stds (nur wenn textboxen sichtbar sind)
        If Not (Controls("TBstd" & CStr(i)).Text = "") Then
            value_anz = value_anz + 1      'Anzahl der Werte (nur wenn textboxen nicht leer sind)
        End If
    End If
Next

'-----
res_mit = 0
For i = 1 To count      '[0-3] 'Summe der Werte bilden
    If Not (Controls("TBstd" & CStr(i)).Text = "") Then
        res_mit = res_mit + CType(Controls("TBstd" & CStr(i)).Text, Double)
    End If
Next
res_mit = res_mit / CType(value_anz, Double)      'Division der Summe durch Anzahl der Werte
Lmit.Text = CType(res_mit, String)
'-----
'Standardabweichung

qu_sum = 0

For i = 1 To count      '[0-3]
    If Not (Controls("TBstd" & CStr(i)).Text = "") Then      'Nur wenn Zelle nicht leer
        qu_sum = qu_sum + ((CType(Controls("TBstd" & CStr(i)).Text, Double) -
            CType(res_mit, Double)) ^ 2)      'Quadratsumme der Differenz bilden
    End If
Next

res_stabw = Math.Sqrt(qu_sum / CType(value_anz - 1, Double))      'Wurzel aus Division der Quadratsumme durch Anzahl d. Werte
res_stabw = res_stabw / res_mit      'Division Stabw durch Mittelwert --> [%]
Lstabw.Text = CType(Math.Round(res_stabw * 100, 2), String)      'Darstellen der Stabw in %

End Sub

Private Sub B_d1_Click(sender As Object, e As EventArgs) Handles B_d1.Click
    'Entfernen des Standardwert 1
    delete_stack.Add(1)      'Füge TB Nummer zu Stapel hinzu (für Rückgängig)
    TBstd1.Text = ""      'Leere TB
    deleted_items(0) = 1      'Setze "zu löschend" Flag
    Call recalc()      'Neuberechnung von Mittelwert und Stabw
    Bundo.Enabled = True      'Aktiviere [Rückgängig]
End Sub

Private Sub B_d2_Click(sender As Object, e As EventArgs) Handles B_d2.Click
    'siehe oben
    delete_stack.Add(2)
    TBstd2.Text = ""
    deleted_items(1) = 1
    Call recalc()
    Bundo.Enabled = True
End Sub

Private Sub B_d3_Click(sender As Object, e As EventArgs) Handles B_d3.Click
    'siehe oben
    delete_stack.Add(3)
    TBstd3.Text = ""
    deleted_items(2) = 1
    Call recalc()
    Bundo.Enabled = True
End Sub

Private Sub B_d4_Click(sender As Object, e As EventArgs) Handles B_d4.Click
    'siehe oben
    delete_stack.Add(4)
    TBstd4.Text = ""
    deleted_items(3) = 1
    Call recalc()
    Bundo.Enabled = True
End Sub

Private Sub B_d5_Click(sender As Object, e As EventArgs) Handles B_d5.Click
    'siehe oben
    delete_stack.Add(5)
    TBstd5.Text = ""
    deleted_items(4) = 1
    Call recalc()
    Bundo.Enabled = True
End Sub

Private Sub B_d6_Click(sender As Object, e As EventArgs) Handles B_d6.Click
    'siehe oben

```

```

delete_stack.Add(6)
TBstd6.Text = ""
deleted_items(5) = 1
Call recalc()
Bundo.Enabled = True
End Sub

Private Sub B_d7_Click(sender As Object, e As EventArgs) Handles B_d7.Click
'siehe oben
delete_stack.Add(7)
TBstd7.Text = ""
deleted_items(6) = 1
Call recalc()
Bundo.Enabled = True
End Sub

Private Sub B_d8_Click(sender As Object, e As EventArgs) Handles B_d8.Click
'siehe oben
delete_stack.Add(8)
deleted_items(7) = 1
Call recalc()
Bundo.Enabled = True
End Sub

Private Sub B_d9_Click(sender As Object, e As EventArgs) Handles B_d9.Click
'siehe oben
delete_stack.Add(9)
TBstd9.Text = ""
deleted_items(8) = 1
Call recalc()
Bundo.Enabled = True
End Sub

Private Sub B_d10_Click(sender As Object, e As EventArgs) Handles B_d10.Click
'siehe oben
delete_stack.Add(10)
TBstd10.Text = ""
deleted_items(9) = 1
Call recalc()
Bundo.Enabled = True
End Sub

End Class

```