



# Fachhochschule Aachen

## Studienort Köln

Fachbereich 9: Medizintechnik und Technomathematik  
Studiengang: Angewandte Mathematik und Informatik

# COBOL

## Projektarbeit

von

**Leon Jarosch**

Matrikelnummer: 3283258

Köln, den 2. Dezember 2023

# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenanalyse</b>	<b>4</b>
1.1	Interpretation der Aufgabe . . . . .	4
1.2	Anforderung an das Programm . . . . .	6
1.3	Fehlerarten . . . . .	6
1.3.1	Technische Fehler . . . . .	6
1.3.2	Syntaktische Fehler . . . . .	7
1.3.3	Semantische Fehler . . . . .	8
1.4	Fehlerbehandlung . . . . .	8
1.4.1	Technische Fehler . . . . .	8
1.4.2	Syntaktische Fehler . . . . .	9
1.4.3	Semantische Fehler . . . . .	9
1.4.4	Sonderfälle . . . . .	9
<b>2</b>	<b>Verfahrensbeschreibung</b>	<b>10</b>
2.1	Gesamtsystem . . . . .	10
2.1.1	Eingabe . . . . .	10
2.1.2	Verarbeitung . . . . .	10
2.1.3	Ausgabe . . . . .	12
2.1.4	Datenstrukturen . . . . .	12
<b>3</b>	<b>Programmbeschreibung</b>	<b>13</b>
3.1	Programmablaufplan . . . . .	13
3.2	Entwicklungsdokumentation . . . . .	13
<b>4</b>	<b>Testdokumentation</b>	<b>19</b>
4.1	Definierte Tests . . . . .	19
4.2	Normalmodus-Tests . . . . .	25
4.3	Explizitmodus-Eingabe-Tests . . . . .	31
4.4	Semantik Explizitmodus-Tests . . . . .	39
4.5	Wortvorschläge-Tests . . . . .	41
4.6	Wörterbücher-Tests . . . . .	42
4.7	Sonderfälle . . . . .	47

<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>49</b>
5.1	Zusammenfassung . . . . .	49
5.2	Ausblick . . . . .	49
<b>A</b>	<b>Benutzeranleitung</b>	<b>51</b>
A.1	Vorbereiten des Systems . . . . .	51
A.1.1	Systemvoraussetzungen . . . . .	51
A.1.2	Installation . . . . .	51
A.2	Programmaufruf . . . . .	51
A.3	Testen der Beispiele . . . . .	51
<b>B</b>	<b>Entwicklungsumgebung</b>	<b>53</b>
<b>C</b>	<b>Verwendete Hilfsmittel</b>	<b>54</b>
<b>D</b>	<b>Erklärung</b>	<b>55</b>
<b>E</b>	<b>Aufgabenstellung</b>	<b>58</b>
<b>F</b>	<b>Quellcode</b>	<b>59</b>

# 1 Aufgabenanalyse

## 1.1 Interpretation der Aufgabe

Im Rahmen des COBOL-Kurses besteht die Aufgabe, ein Programm zu entwickeln, welches Abrechnungsdaten aus einem Journal einliest und auswertet.

Das Journal ist eine Textdatei, die Zeilen weise Abrechnungen auflistet. Teil einer Abrechnung sind:

Die Datei ist nach KundenID und danach nach LeistungsID vorsortiert.

Für jeden erkannten Kunden soll nun eine Rechnung erstellt werden, welche auswertet welche Leistungen in anspruch genommen wurden und wie viel diese gekostet haben.

Abschließend soll die Gesamtabrechnung errechnet werden, welche die Gesamtkosten eines Kunden zusammenfasst.

Die Rechnungen aller Kunden sollen in einer Rechnungsdatei gemeinsam abgespeichert werden.

Es soll eine Eingabemethode entwickelt werden, die pro Buchstabe nur eine Zifferntaste benötigt, indem für alle Buchstaben eines Worts die Taste, in der sie vorkommen, lediglich einmal gedrückt wird. START würde bei dieser Methode mit dem Zifferncode 78278 kodiert werden. Hierzu soll das Programm ein Wörterbuch anlegen, welches ein nicht injektives Mapping von Zifferncodes zu Wörtern und deren Häufigkeit speichert. Ist ein Wort noch nicht im Wörterbuch bekannt, kann es über einen expliziten Modus eingegeben werden. Dieser Modus nimmt ebenfalls Zifferncodes entgegen, mit der Besonderheit, dass für jeden Buchstaben des Wortes, welches eingegeben werden soll auch die Anzahl der Tastendrucke angegeben wird, die nötig sind, um den jeweiligen Buchstaben zu codieren.

START würde im expliziten Modus mit 7481217381 codiert werden, wobei die Werte an den geraden Indizes (beginnend bei 1) für die Anzahl der Tastendrucke stehen. Das Abschließen der Eingabe eines Wortes soll über ein Leerzeichen (Taste 1) oder einen Punkt (Taste 0) erfolgen. Das gilt sowohl für den expliziten Modus, als auch für den normalen Modus. Nach vollständiger Eingabe im expliziten Modus wird das Wort im Wörterbuch abgespeichert.

Wie oben erwähnt ist das Mapping nicht injektiv, weshalb nach jeder Eingabe eines Wortes eine Bestätigung des Nutzers erfolgen soll. Akzeptiert er das Wort, soll die Häufigkeit dieses Wortes um eins erhöht werden.

Es gibt genau drei Fälle einer Eingabe im normalen Modus, die abgedeckt werden müssen:

- Fall 1:** Es existiert bisher kein Wort zum eingegebenen Zifferncode. Der Nutzer wird in den Explizitmodus weitergeleitet.
- Fall 2:** Es ist genau ein Wort zu diesem Code im Wörterbuch vorhanden. Der Nutzer kann das Wort akzeptieren und es wird Teil des zu bildenden Satzes oder ablehnen und in den expliziten Modus wechseln.
- Fall 3:** Es sind mehrere Wörter im Wörterbuch zum Code vorhanden. Es werden der Reihe nach alle Wörter vorgeschlagen, sortiert nach der Häufigkeit.

Zudem soll die Möglichkeit bestehen, ein bereits vorhandenes Wörterbuch, welches beispielsweise aus früheren Programmaufrufen entstanden ist, einlesen zu können. Dieses Wörterbuch soll mit einem Texteditor bearbeitbar sein und einem Schema entsprechen. Dabei soll ein Wörterbucheintrag einer Zeile entsprechen und den Aufbau [Zahlencode Wort Häufigkeit] haben. Die einzelnen Komponenten werden demnach mit einem Leerzeichen getrennt.

Weitergehend gibt es Beschränkungen, welche einzuhalten sind:

- Die maximale Wortlänge betrage 15 Zeichen, die maximale Satzlänge 200 Zeichen.
- Die maximale Anzahl an Einträgen im Wörterbuch kann begrenzt werden, soll aber durch eine simple Änderung im Source Code anpassbar sein.

## 1.2 Anforderung an das Programm

Aus der Aufgabenstellung geht hervor, dass das Programm folgenden Anforderungen genügen muss:

Es muss

- Wörter über die Tastatur einlesen
- Sätze bilden und ausgeben
- ein Wörterbuch anlegen und erweitern
- vorhandene Wörterbücher einlesen
- das Wörterbuch beim Exportieren sortieren
- die Wortvorschläge sortiert nach der Häufigkeit anzeigen

können. Zusätzlich sollte das Programm eine angemessene Laufzeit haben und geeignete Datenstrukturen verwenden.

## 1.3 Fehlerarten

Während der Laufzeit des Programms kann es zu unterschiedlichen Fehlerarten kommen, die alle eine passende Behandlung brauchen.

### 1.3.1 Technische Fehler

Durch das Arbeiten mit Dateien kann es zu technischen Fehlern kommen. Wenn der Nutzer zum Beispiel ein externes Wörterbuch einlesen möchte und es im falschen Verzeichnis abgelegt hat, muss das Programm entsprechend reagieren.

Das Programm muss außerdem mit unerwarteten Benutzer-Eingaben umgehen können. Ist das nicht der Fall, könnte sich der Benutzer-Dialog in einer Endlosschleife aufhängen.

### 1.3.2 Syntaktische Fehler

Durch die vielen Benutzereingaben existieren sehr viele Stellen, an denen es zu syntaktischen Fehlern kommen kann. Um Fehler zu vermeiden, muss ein externes Wörterbuch den folgenden Anforderungen entsprechen:

- Einzelne Wörterbucheinträge werden zeilenweise gespeichert.
- Ein Eintrag besteht aus drei Komponenten: Das T9-Kodierte Wort (max. 15 Ziffern), dem Wort-Text (max. 15 Zeichen) und der Häufigkeit (max. dreistellige Ganzzahl).
- Die drei Komponenten werden jeweils durch ein Leerzeichen getrennt.

Werden diese Bedingungen nicht eingehalten, wird es zu Fehlern kommen. Bei der Auswahl des Wörterbuchs kann der Nutzer außerdem den Namen falsch eingeben, was wiederum Probleme verursacht.

Des Weiteren könnte der Nutzer bei der Eingabe im normalen Modus und im Explizitmodus die syntaktischen Vorgaben missachten und damit Fehler hervorrufen. Für die Eingabe im normalen Modus gilt:

- Es sind maximal 16 Zeichen erlaubt (15 Buchstaben und 1 Punkt bzw. Leerzeichen).
- Es müssen mindestens 2 Zeichen eingegeben werden.
- Außer den Ziffern von 0-9 und dem Leerzeichen sind keine anderen Zeichen erlaubt.
- Es muss genau eine 1 (Leerzeichen) oder eine 0 (Punkt) am Wortende stehen. An anderen Stellen im Wort sind diese beiden Zeichen nicht erlaubt.
- Innerhalb des Wortes sind keine Leerzeichen (□) erlaubt.
- Nach einer 0 oder 1 dürfen nur noch Leerzeichen stehen.

Für die Eingabe im Explizitmodus gelten ähnliche Vorgaben:

- Es sind maximal 31 Zeichen erlaubt (15 Buchstaben, kodiert als Paare, und 1 Punkt bzw. Leerzeichen).

- Es müssen mindestens 3 Zeichen eingegeben werden.
- Außer den Ziffern von 0-9 und dem Leerzeichen sind keine anderen Zeichen erlaubt.
- Es muss genau eine 1 (Leerzeichen) oder eine 0 (Punkt) am Wortende stehen.
- Innerhalb des Wortes sind keine Leerzeichen erlaubt.
- Nach dem abschließenden Satzzeichen dürfen nur noch Leerzeichen stehen.

Das Bestätigen oder Ablehnen eines Vorschlags wird im Programm durch \* (Ja) und # (Nein) kodiert. Wenn der Anwender hier etwas anderes eingibt, können auch Fehler auftreten.

### 1.3.3 Semantische Fehler

Es existieren Eingaben, die den oben erwähnten syntaktischen Regeln genügen, im Programm aber trotzdem Fehler verursachen. Wenn im Explizitmodus zum Beispiel die Eingabe „25“ getätigt wird, kann dazu kein passender Buchstabe gefunden werden, weil die Eingabe nicht dem strukturellen Aufbau der Tastatur entspricht. Gleiches gilt für die Eingabe „24“. Die Taste 2 hat nur 3 mögliche Buchstaben.

## 1.4 Fehlerbehandlung

Die unterschiedlichen Fehlerarten müssen alle erkannt und im Anschluss sinnvoll behandelt werden.

### 1.4.1 Technische Fehler

Wenn ein, vom Anwender ausgewähltes, Wörterbuch nicht gefunden wird, gibt es erneut die Möglichkeit ein Wörterbuch auszuwählen. Es kann aber auch weitergehen, ohne, dass ein externes Wörterbuch ausgewählt wurde.

Auch bei unerwarteten Eingaben wird eine erneute Eingabe gefordert, bis eine erwartete Eingabe getätigt wird. So wird gesichert, dass sich das Programm deterministisch verhält und nie in einer Endlosschleife landet.



### **1.4.2 Syntaktische Fehler**

Beim Einlesen des Wörterbuchs wird für jeden Eintrag geprüft, ob die gewünschte Struktur eingehalten wurde. Ist das mindestens ein mal nicht der Fall, wird das Wörterbuch abgelehnt. Der Anwender bekommt dann die Möglichkeit ein anderes Wörterbuch einzulesen oder mit einem leeren Wörterbuch fortzufahren.

Jede Benutzereingabe wird auf die oben definierten syntaktischen Regeln geprüft. Auch hier gilt: Die Eingabe wird so lange wiederholt, bis die Regeln eingehalten werden.

### **1.4.3 Semantische Fehler**

Entspricht ein eingegebener Explizit-Code nicht dem strukturellen Aufbau der Tastatur, wird eine passende Fehlermeldung ausgegeben und eine erneute Eingabe gefordert.

### **1.4.4 Sonderfälle**

Ein in der Aufgabenstellung nicht angesprochener Sonderfall tritt auf, wenn der Anwender im normalen Modus ein nicht im Wörterbuch enthaltenes Wort eingibt, in den Explizitmodus weitergeleitet wird und dort ein komplett anderes Wort eingibt. Zum Beispiel kann im normalen Modus „3370“ eingegeben werden. Das Wort wird anschließend nicht im Wörterbuch gefunden und der Anwender gibt im Explizitmodus „33437423420“ ein. Der Anwender wird in dieser Situation lediglich durch eine Ausgabe auf den Fehler aufmerksam gemacht. Wenn der zur Explizit-Eingabe passende Code „34724“ schon im Wörterbuch enthalten ist, wird seine Häufigkeit inkrementiert. Ist das nicht der Fall, wird er als neuer Eintrag im Wörterbuch aufgenommen.

## 2 Verfahrensbeschreibung

### 2.1 Gesamtsystem

Das System arbeitet nach dem **E**ingabe, **V**erarbeitung, **A**usgabe-Prinzip, kurz EVA. EVA ist ein Grundprinzip der Datenverarbeitung, bei welchem die drei Schritte sequenziell durchlaufen werden. Das Prinzip ist im Normalfall zustandslos, weshalb dieses System nicht dem reinen EVA-Prinzip entspricht. Es ist mehr eine leichte Abwandlung dessen, da Eingaben, zu unterschiedlichen Zeitpunkten, zu einer anderen Ausgabe führen können.

#### 2.1.1 Eingabe

Es gibt zwei unterschiedliche Eingabemodi. Sowohl der Explizit- als auch der Normalmodus nehmen über die Kommandozeile Eingaben entgegen. Der Normalmodus ist für die Eingabe von bereits bekannten Wörtern im Wörterbuch gedacht, wohingegen der Explizitmodus neue, unbekannte Wörter entgegennimmt. Einen Spezialfall bietet das Einlesen eines bereits vorhandenen Wörterbuchs. Es lässt sich zu Beginn des Programmstarts ein Wörterbuch einlesen, um das vom Programm genutzte Wörterbuch zu befüllen. Der Nutzer muss das externe Wörterbuch am besten in den Ordner legen, in welchem sich die .exe-Datei befindet und beim Start des Programms den Namen der Datei angeben. Ansonsten kann der Nutzer auch einen relativen bzw. absoluten Pfad angeben.

#### 2.1.2 Verarbeitung

Je nach Eingabemodus wird die Nutzereingabe unterschiedlich verarbeitet. Ist sie über den Normalmodus passiert, wird im Wörterbuch ein passendes Mapping

zum eingegebenen Zahlencode gesucht. Je nach den zuvor definierten Fällen wird unterschiedlich vorgegangen.

Im Fall 1 wird der Nutzer automatisch in den Explizitmodus geleitet.

Im Fall 2 wird der Nutzer durch Eingabeaufforderung gefragt, ob das gefundene Wort gewünscht ist. Falls *Ja* wird das Wort dem aktuellen Satz hinzugefügt, falls *Nein* wird der Nutzer automatisch in den Explizitmodus geleitet.

Im letzten Fall 3 wird für jedes der gefundenen Wörter abgefragt, ob es das gewünschte ist, bis der Nutzer das erste Mal mit einem *Ja* antwortet. Sollte es keine passenden Wörter mehr geben, da der Nutzer durchgehend mit *Nein* geantwortet hat, wird der Nutzer auch hier in den Explizitmodus geleitet.

Erfolgte die Eingabe über den Explizitmodus wird zuerst überprüft, ob sie legal ist. Z. B. dürfen keine Symbole, sondern nur Ziffern, eingegeben werden und die Eingabe muss mit einem  $\square$  (Taste 1) oder  $.$  (Taste 0) enden. Siehe hierzu Syntaktische Fehler.

Außerdem erkennt das Programm Unregelmäßigkeiten und informiert den Nutzer z. B. darüber, dass die Eingabe, die er zuvor im Normalmodus getätigt hat, nicht mit dem eingegebenen Zahlencode im Explizitmodus übereinstimmt. In diesem speziellen Fall wird das in expliziter Form eingegebene Wort trotzdem gespeichert. Ist die Eingabe korrekt und legal, wird noch überprüft, ob das eingegebene Wort schon im Wörterbuch existiert und falls dies der Fall ist, lediglich die Häufigkeit um eins erhöht. Sollten die Überprüfungen alle zusagen, wird das Wort im Wörterbuch gespeichert und der Nutzer gelangt wieder in den Normalmodus.

Das Abspeichern im Wörterbuch folgt einem Schema, wobei jedes Wort in einer .txt-Datei eine Zeile erhält. Jede Zeile besteht aus drei Spalten, getrennt mit einem Leerzeichen. Die erste Spalte enthält den Zifferncode für das Wort, die zweite das Wort selbst und die dritte Spalte die Häufigkeit des Worts.

Der Eintrag HUT sieht in der .txt-Datei z. B. wie folgt aus:

488 HUT 1

### 2.1.3 Ausgabe

Ausgaben erscheinen ebenfalls in dem Fenster der Kommandozeile und informieren den Nutzer über die Verarbeitung seiner Eingaben. Sollte er beispielsweise einen Satz beendet haben, wird der ganze Satz angezeigt. Auch Fehlermeldungen werden in der Konsole angezeigt, um den Nutzer auf falsche Eingaben oder Verarbeitung hinzuweisen. Das angelegte Wörterbuch ist ebenfalls eine Form der Ausgabe und kann im bin/ Verzeichnis eingesehen werden.

### 2.1.4 Datenstrukturen

Zum effizienten Arbeiten mit dem Wörterbuch wird die Eingabedatei in eine COBOL-Tabelle geschrieben. Diese wird mit einem Spezialindex definiert, um das Verwenden der „SEARCH“-Funktion zu ermöglichen. Am Programmende wird die Tabelle dann wieder in eine Datei geschrieben.

Auch das Tastenfeld zur Expliziteingabe wird als Tabelle realisiert. Jeder Tabelleneintrag repräsentiert dabei eine Taste. So kann mit zwei Indizes elegant auf einzelne Buchstaben zugegriffen werden.

Im gesamten Programm wird hauptsächlich mit alphanumerischen Feldern gearbeitet. Dadurch können dem Anwender sehr ausführliche Fehlermeldungen angezeigt werden.

## 3 Programmbeschreibung

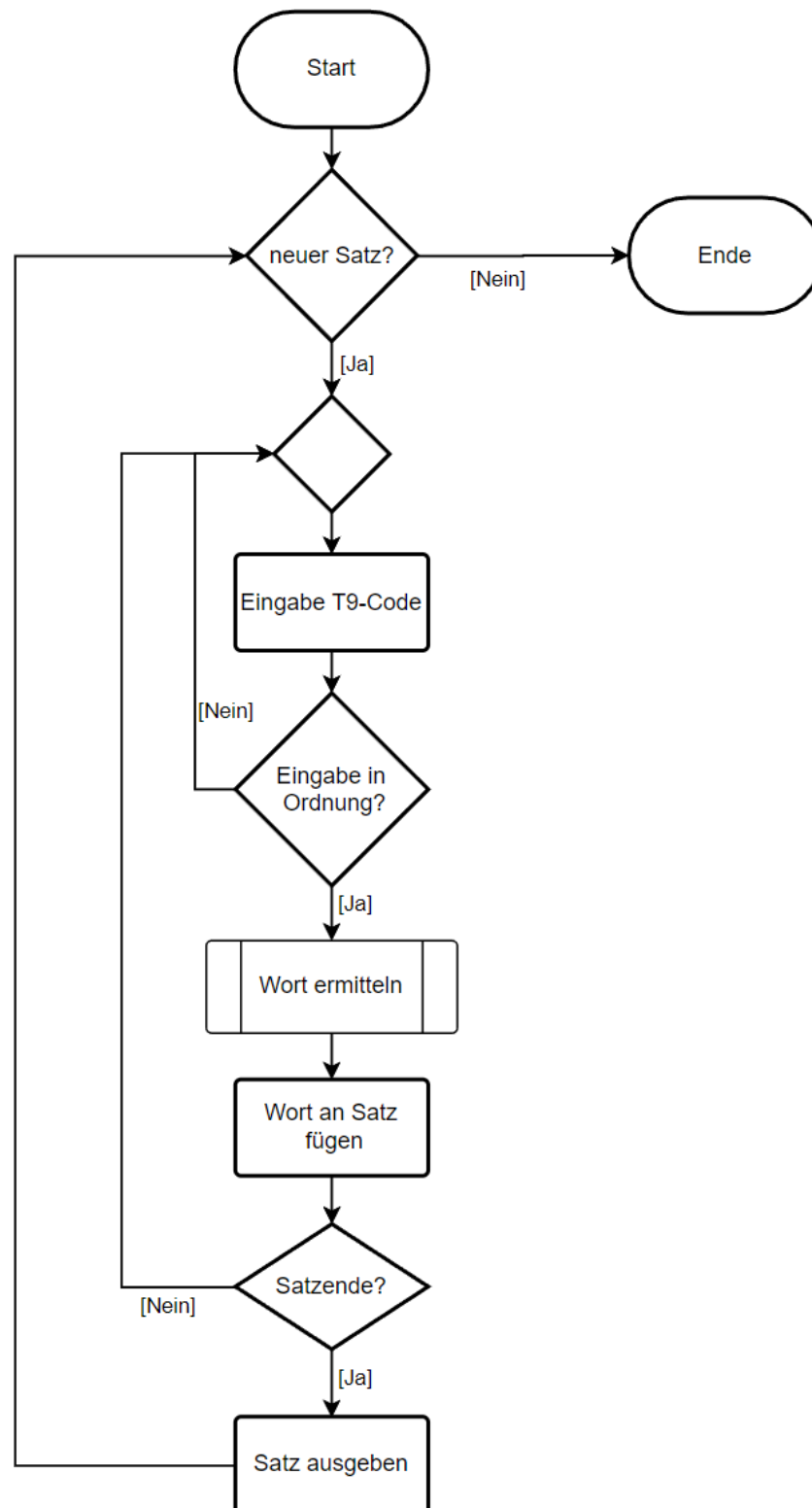
### 3.1 Programmablaufplan

Der Ablauf des Programms ist sequenziell und kann daher gut mit Programmablaufplänen dargestellt werden. Die folgenden Abbildungen beschreiben Teile des Programms. Abbildung 3.1 zeigt, wie aus wiederholten Benutzereingaben Sätze konstruiert werden. Die Auswahl eines externen Wörterbuchs wird in Abbildung 3.2 beschrieben. In Abbildung 3.3 wird der Ablauf des Explizimodus visualisiert. Die Ermittlung eines Wortes, aus einem eingegebenen T9-Code, kann in Abbildung 3.4 beobachtet werden.

### 3.2 Entwicklungsdokumentation

Es wurden grundsätzlich sprechende Namen für Variablen, Abschnitte und Paragraphen gewählt. Daher bedarf es nur geringer Dokumentation. Die Funktionen der einzelnen Paragraphen sind in Tabelle 3.1 beschrieben.

Wenn Variablen zu einer bestimmten logischen Einheit gehören, haben sie einen passenden Präfix.



**Abbildung 3.1:** Benutzerdialog: Konstruieren von Sätzen aus Sicht des Anwenders.

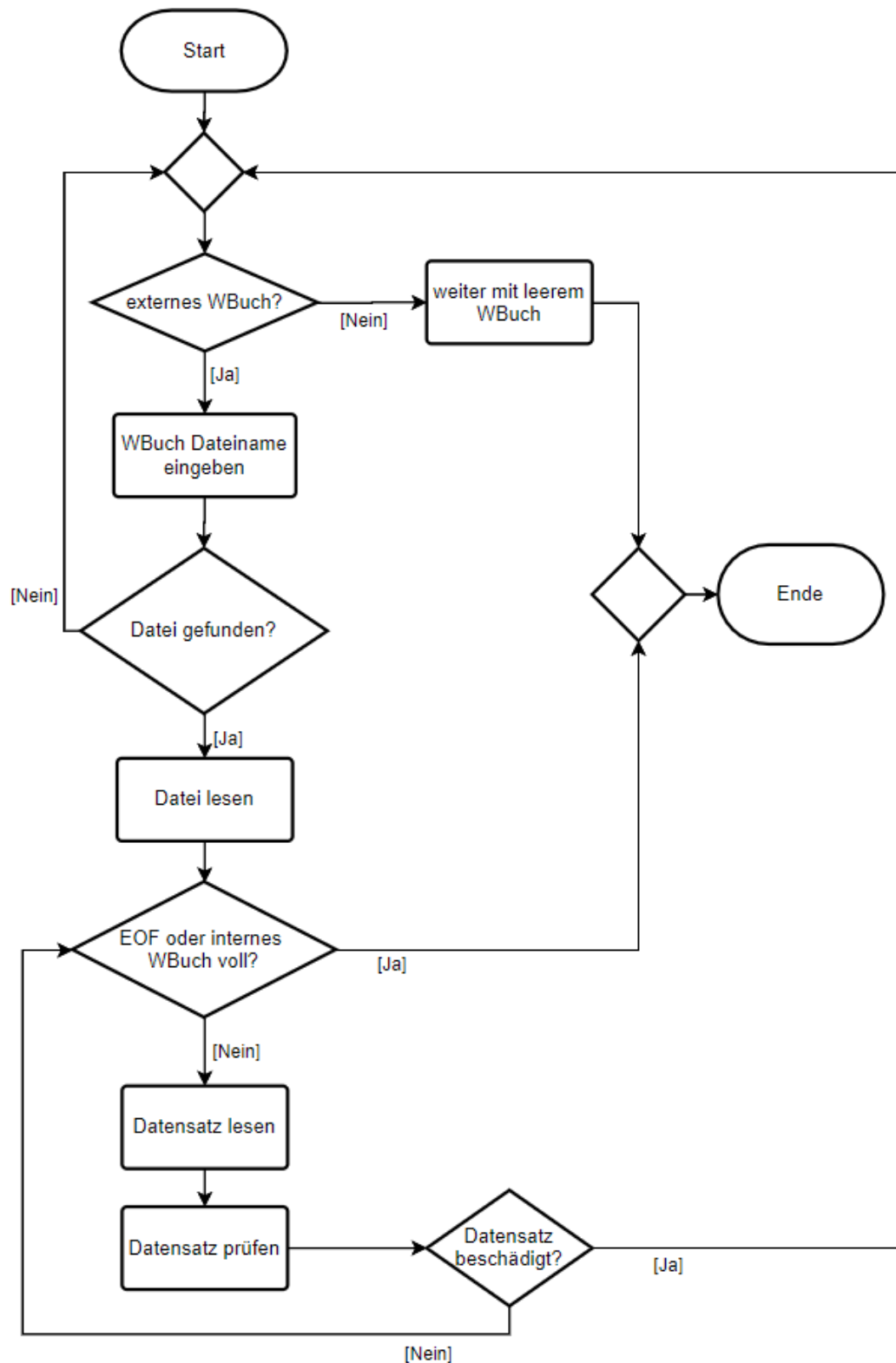


Abbildung 3.2: Einlesen eines externen Wörterbuchs.

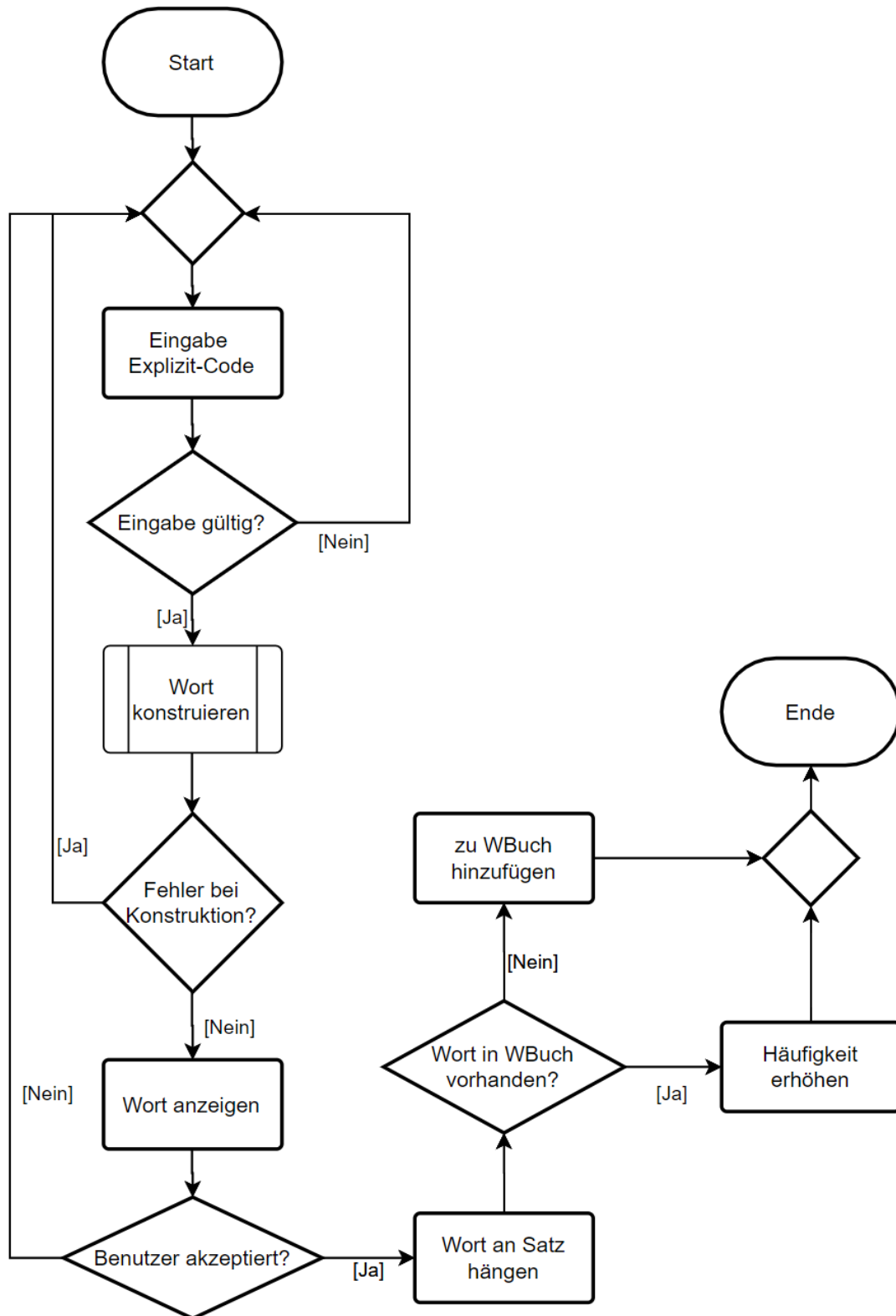


Abbildung 3.3: Ablauf des Explizitmodus.



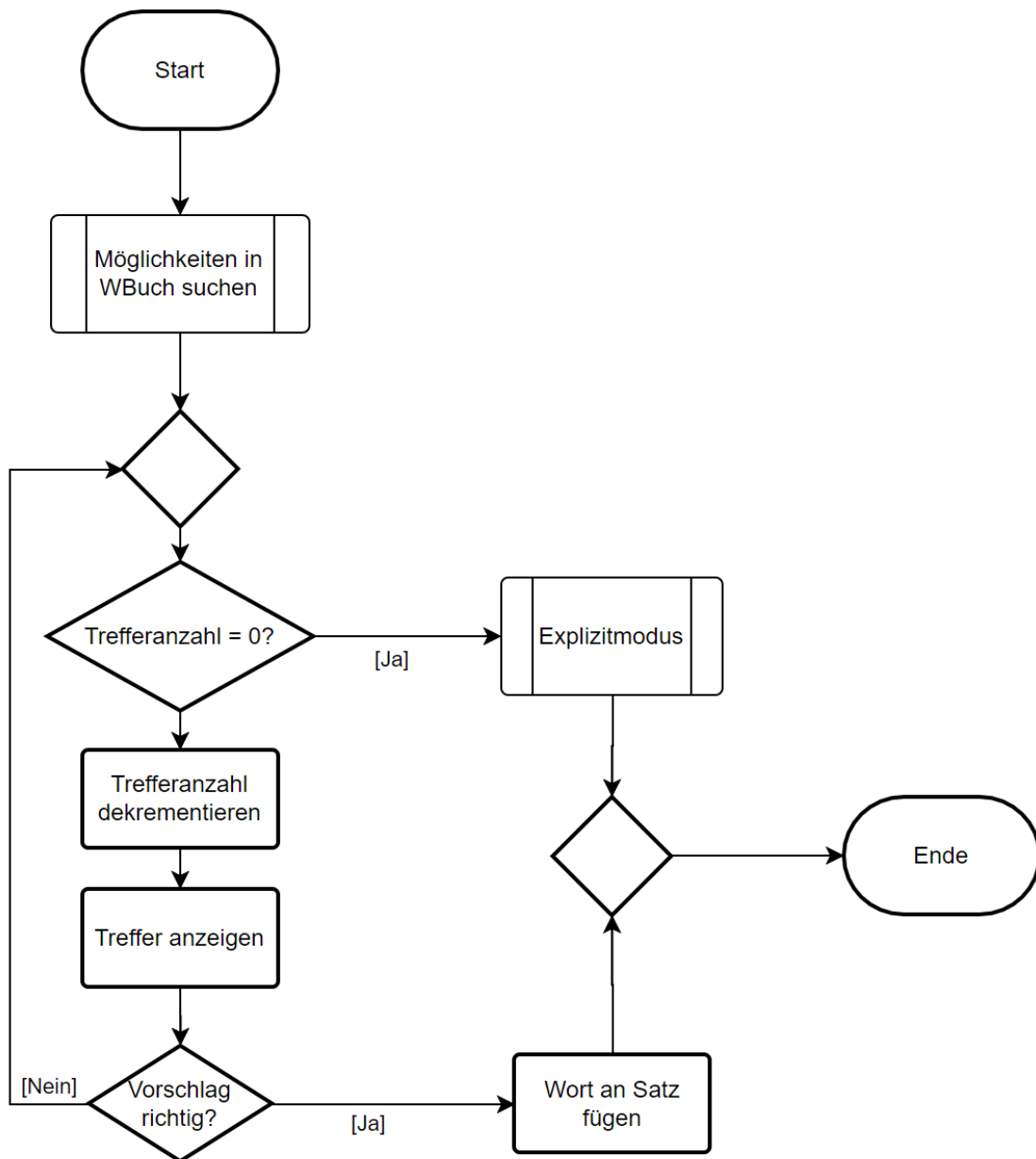


Abbildung 3.4: Ermitteln eines Wortes.

Bezeichnung	Beschreibung
MAIN-PROCEDURE	Hauptablauf, in dem einzelne Aufgaben an andere Paragraphen delegiert werden
Vorlauf SECTION	Abschnitt zur Vorbereitung des Programms
Auswahl-WBuch	Es wird die Auswahl eines externen Wörterbuchs angeboten
Einlesen-WBuch	Paragraf zum Einlesen eines, zuvor spezifizierten, Wörterbuchs
Lese-Satz	Verarbeitungsschritt einer einzelnen Zeile eines externen Wörterbuchs
Init-Explizit-Tab	Initialisierung des strukturellen Aufbaus der Handytastatur
Hauptlauf SECTION	Abschnitt zur Hauptaufgabe des Programms
Benutzer-Dialog	Paragraf zum Steuern des Benutzer-Dialogs
Ermittle-Wort	Ermittelt ein Wort aus einem eingegebenen T9-Code
Wort-Auswahl	Paragraf schlägt passende Wörter zum eingegebenen T9-Code vor, und ermöglicht eine Auswahl
Finde-Moeglichkeiten	Es werden alle, zum eingegebenen Code passenden, Wörter im Wörterbuch gefunden und zwischengespeichert
Sortiere-Nach-Haeuf	Bubble-Sort zum Sortieren der zwischengespeicherten Treffer
Explizit-Eingabe	Steuerung der Eingabe im Explizitmodus
Suche-Wort-In-WBuch	Prüft, ob ein Wort bereits im Wörterbuch vorhanden ist
Konstruiere-Wort	Konstruiert ein Wort aus einem eingegebenen Explizit-Code
Pruefe-Explizit-Eingabe	Validiert eine Benutzereingabe im Explizitmodus
Pruefe-Eingabe	Validiert eine Benutzereingabe im normalen Modus
Nachlauf SECTION	Abschnitt zum Speichern des Wörterbuchs
Schreibe-WBuch-Sortiert	Sortiert das Wörterbuch alphabetisch und speichert es in einer Datei

**Tabelle 3.1:** Aufgaben der einzelnen logischen Einheiten.

## 4 Testdokumentation

Alle Testfälle können wie beschrieben in „Testen der Beispiele“ ausgeführt werden. Für eine klare Struktur wurden Sie in 5 verschiedene Testgruppen eingeteilt:

**Gruppe 1:** Tests, welche aus der Aufgabenstellung hervorgehen.

**Gruppe 2:** Überprüfungen der Eingabe im T9- bzw. Normalmodus.

**Gruppe 3:** Überprüfungen der Eingabe im Explizitmodus.

**Gruppe 4:** Tests, welche speziell auf die Semantik im Explizitmodus eingehen.

**Gruppe 5:** Tests, welche die Reihenfolge der Wortvorschläge überprüfen

**Gruppe 6:** Tests, welche das Einlesen und Ausgeben von Wörterbüchern überprüfen.

**Gruppe 7:** Tests, zum Überprüfen der beschriebenen Sonderfälle (Kapitel 1.4.4)

### 4.1 Definierte Tests

Die in der Aufgabenstellung definierten Tests werden im Folgenden ausführlich beschrieben. Dabei werden die allgemeine Funktionalität des Programms getestet und Normalfälle von vielen Programmkomponenten abgedeckt.

#### Normalfälle

**T1.1** Beginnend mit einem Leeren Wörterbuch wird getestet, ob der Satz „DER HUT IST EIN FES.“ gebildet werden kann. Wörterbuch:

```
1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /  
   nein (#)  
2 #  
3 Kein Woerterbuch geladen.
```

```
4 Beginnen Sie einen neuen Satz oder beenden Sie das Programm
   mit (0).
5 Bitte geben Sie ein kodierte Wort ein:
6 3371
7 Im Woerterbuch wurde keine passender Eintrag gefunden!
8 Bitte Wort im Explizit-Modus eingeben:
9 3132731
10 DER
11 Wort in Ordnung? ja (*) / nein (#)
12 *
13 Das Wort DER wurde mit Code 337 im Woerterbuch
   abgespeichert!
14 Bitte geben Sie ein kodierte Wort ein:
15 4881
16 Im Woerterbuch wurde keine passender Eintrag gefunden!
17 Bitte Wort im Explizit-Modus eingeben:
18 4282811
19 HUT
20 Wort in Ordnung? ja (*) / nein (#)
21 *
22 Das Wort HUT wurde mit Code 488 im Woerterbuch
   abgespeichert!
23 Bitte geben Sie ein kodierte Wort ein:
24 4781
25 Im Woerterbuch wurde keine passender Eintrag gefunden!
26 Bitte Wort im Explizit-Modus eingeben:
27 4374811
28 IST
29 Wort in Ordnung? ja (*) / nein (#)
30 *
31 Das Wort IST wurde mit Code 478 im Woerterbuch
   abgespeichert!
32 Bitte geben Sie ein kodierte Wort ein:
```

```
33 3461
34 Im Woerterbuch wurde keine passender Eintrag gefunden!
35 Bitte Wort im Explizit-Modus eingeben:
36 3243621
37 EIN
38 Wort in Ordnung? ja (*) / nein (#)
39 *
40 Das Wort EIN wurde mit Code 346 im Woerterbuch
   abgespeichert!
41 Bitte geben Sie ein kodierte Wort ein:
42 3370
43 DER
44 Wort in Ordnung? ja (*) / nein (#)
45 #
46 Bitte Wort im Explizit-Modus eingeben:
47 3332740
48 FES
49 Wort in Ordnung? ja (*) / nein (#)
50 *
51 Das Wort FES wurde mit Code 337 im Woerterbuch
   abgespeichert!
52 Eingebener Satz:
53 DER HUT IST EIN FES.
54 Beginnen Sie einen neuen Satz oder beenden Sie das Programm
   mit (0).
55 Bitte geben Sie ein kodierte Wort ein:
56 0
57 Programm Ende.
```

337 DER 1
346 EIN 1
337 FES 1
488 HUT 1
478 IST 1

Abbildung 4.1: Wörterbuch nach Test 1.1.

**T1.2** Es sollen die beiden Sätze „DER SATZ IST KURZ. EIN FES IST EIN HUT.“ konstruiert werden. Dabei wird das Wörterbuch 4.1 aus dem vorherigen Test verwendet.

```

1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /
  nein (#)
2 *
3 Bitte geben sie den Dateinamen des Woerterbuchs ein.
4 Woerterbuch-out.txt
5 Woerterbuch erfolgreich eingelesen.
6 Beginnen Sie einen neuen Satz oder beenden Sie das Programm
  mit (0).
7 Bitte geben Sie ein kodierte Wort ein:
8 3371
9 DER
10 Wort in Ordnung? ja (*) / nein (#)
11 *
12 Bitte geben Sie ein kodierte Wort ein:
13 72891
14 Im Woerterbuch wurde keine passender Eintrag gefunden!
15 Bitte Wort im Explizit-Modus eingeben:
16 742181941
17 SATZ
18 Wort in Ordnung? ja (*) / nein (#)
19 *

```

```
20 Das Wort SATZ wurde mit Code 7289 im Woerterbuch
    abgespeichert!
21 Bitte geben Sie ein kodierte Wort ein:
22 4781
23 IST
24 Wort in Ordnung? ja (*) / nein (#)
25 *
26 Bitte geben Sie ein kodierte Wort ein:
27 58790
28 Im Woerterbuch wurde keine passender Eintrag gefunden!
29 Bitte Wort im Explizit-Modus eingeben:
30 528273940
31 KURZ
32 Wort in Ordnung? ja (*) / nein (#)
33 *
34 Das Wort KURZ wurde mit Code 5879 im Woerterbuch
    abgespeichert!
35 Eingebener Satz:
36 DER SATZ IST KURZ.
37 Beginnen Sie einen neuen Satz oder beenden Sie das Programm
    mit (0).
38 Bitte geben Sie ein kodierte Wort ein:
39 3461
40 EIN
41 Wort in Ordnung? ja (*) / nein (#)
42 *
43 Bitte geben Sie ein kodierte Wort ein:
44 3371
45 DER
46 Wort in Ordnung? ja (*) / nein (#)
47 #
48 FES
49 Wort in Ordnung? ja (*) / nein (#)
```

```
50 *
51 Bitte geben Sie ein kodierte Wort ein:
52 4781
53 IST
54 Wort in Ordnung? ja (*) / nein (#)
55 *
56 Bitte geben Sie ein kodierte Wort ein:
57 3461
58 EIN
59 Wort in Ordnung? ja (*) / nein (#)
60 *
61 Bitte geben Sie ein kodierte Wort ein:
62 4880
63 HUT
64 Wort in Ordnung? ja (*) / nein (#)
65 *
66 Eingebener Satz:
67 EIN FES IST EIN HUT.
68 Beginnen Sie einen neuen Satz oder beenden Sie das Programm
    mit (0).
69 Bitte geben Sie ein kodierte Wort ein:
70 0
71 Programm Ende.
```

337 DER 2
346 EIN 3
337 FES 2
488 HUT 2
478 IST 3
5879 KURZ 1
7289 SATZ 1

**Abbildung 4.2:** Wörterbuch nach Test 1.2.



## 4.2 Normalmodus-Tests

In den folgenden Testfällen wird geprüft, ob das Programm bei Verletzung der in Kapitel 1.3.2 syntaktischen Regeln für den normalen Modus passend reagiert. In allen Fällen sollte ein Fehler gemeldet und eine neue Eingabe gefordert werden.

### Fehlerfälle

#### T2.1 Leerzeichen im Wort.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76 3892730
- 3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 4 Bitte geben Sie ein kodierte Wort ein:

#### T2.2 Ungültiges Wortende.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76389273
- 3 Es muss genau ein Leerzeichen (1) oder Punkt (0) am Wortende stehen!
- 4 Bitte geben Sie ein kodierte Wort ein:

#### T2.3 Unerlaubte Zeichen-Eingabe.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76389g2a730
- 3 Die Eingabe darf nur Ziffern von 0–9 enthalten!
- 4 Bitte geben Sie ein kodierte Wort ein:

#### T2.4 Zeichen nach Wortende.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 763890273
- 3 Nach dem Leerzeichen bzw. Punkt dürfen keine weiteren Zeichen folgen!

4 Bitte geben Sie ein kodiertes Wort ein:

**T2.5** Mehr als ein Wortende.

1 Bitte geben Sie ein kodiertes Wort ein:

2 7638927310

3 Es muss genau ein Leerzeichen (1) oder Punkt (0) am Wortende stehen!

4 Bitte geben Sie ein kodiertes Wort ein:

**T2.6** Leerzeichen im Wort und ungültiges Ende.

1 Bitte geben Sie ein kodiertes Wort ein:

2 76 389273

3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!

4 Es muss genau ein Leerzeichen (1) oder Punkt (0) am Wortende stehen!

5 Bitte geben Sie ein kodiertes Wort ein:

**T2.7** Leerzeichen im Wort und unerlaubte Zeichen.

1 Bitte geben Sie ein kodiertes Wort ein:

2 763 89g2a730

3 Die Eingabe darf nur Ziffern von 0–9 enthalten!

4 Leerzeichen innerhalb des Wortes sind nicht erlaubt!

5 Bitte geben Sie ein kodiertes Wort ein:

**T2.8** Leerzeichen im Wort und Zeichen nach Wortende.

1 Bitte geben Sie ein kodiertes Wort ein:

2 7638902 73

3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!

4 Nach dem Leerzeichen bzw. Punkt dürfen keine weiteren Zeichen folgen!

5 Bitte geben Sie ein kodiertes Wort ein:

**T2.9** Leerzeichen im Wort und mehrfaches Wortende.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76389 27310
- 3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 4 Es muss genau ein Leerzeichen (1) oder Punkt (0) am Wortende stehen!
- 5 Bitte geben Sie ein kodierte Wort ein:

**T2.10** Unerlaubte Zeichen und fehlendes Wortende.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76389g2a73
- 3 Die Eingabe darf nur Ziffern von 0–9 enthalten!
- 4 Es muss genau ein Leerzeichen (1) oder Punkt (0) am Wortende stehen!
- 5 Bitte geben Sie ein kodierte Wort ein:

**T2.11** Unerlaubte Zeichen und Zeichen nach Wortende.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76389g2a703
- 3 Die Eingabe darf nur Ziffern von 0–9 enthalten!
- 4 Nach dem Leerzeichen bzw. Punkt dürfen keine weiteren Zeichen folgen!
- 5 Bitte geben Sie ein kodierte Wort ein:

**T2.12** Unerlaubte Zeichen und mehrfaches Wortende.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76389g2a7310
- 3 Die Eingabe darf nur Ziffern von 0–9 enthalten!
- 4 Es muss genau ein Leerzeichen (1) oder Punkt (0) am Wortende stehen!
- 5 Bitte geben Sie ein kodierte Wort ein:

**T2.13** Mehrfaches Wortende und Zeichen nach Wortende.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76389027310
- 3 Es muss genau ein Leerzeichen (1) oder Punkt (0) am  
Wortende stehen!
- 4 Bitte geben Sie ein kodierte Wort ein:

**T2.14** Leerzeichen im Wort und unerlaubte Zeichen und fehlendes Wortende.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76389g2 a73
- 3 Die Eingabe darf nur Ziffern von 0–9 enthalten!
- 4 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 5 Es muss genau ein Leerzeichen (1) oder Punkt (0) am  
Wortende stehen!
- 6 Bitte geben Sie ein kodierte Wort ein:

**T2.15** Alle vorherigen Fehler nacheinander und anschliessend korrekte Eingabe.

- 1 Bitte geben Sie ein kodierte Wort ein:
- 2 76 3892730
- 3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 4 Bitte geben Sie ein kodierte Wort ein:
- 5 76389273
- 6 Es muss genau ein Leerzeichen (1) oder Punkt (0) am  
Wortende stehen!
- 7 Bitte geben Sie ein kodierte Wort ein:
- 8 76389g2a730
- 9 Die Eingabe darf nur Ziffern von 0–9 enthalten!
- 10 Bitte geben Sie ein kodierte Wort ein:
- 11 763890273
- 12 Nach dem Leerzeichen bzw. Punkt dürfen keine weiteren  
Zeichen folgen!
- 13 Bitte geben Sie ein kodierte Wort ein:

14 7638927310  
15 Es muss genau ein Leerzeichen (1) oder Punkt (0) am  
Wortende stehen!  
16 Bitte geben Sie ein kodierte Wort ein:  
17 76 389273  
18 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
19 Es muss genau ein Leerzeichen (1) oder Punkt (0) am  
Wortende stehen!  
20 Bitte geben Sie ein kodierte Wort ein:  
21 763 89g2a730  
22 Die Eingabe darf nur Ziffern von 0–9 enthalten!  
23 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
24 Bitte geben Sie ein kodierte Wort ein:  
25 7638902 73  
26 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
27 Nach dem Leerzeichen bzw. Punkt duerfen keine weiteren  
Zeichen folgen!  
28 Bitte geben Sie ein kodierte Wort ein:  
29 76389 27310  
30 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
31 Es muss genau ein Leerzeichen (1) oder Punkt (0) am  
Wortende stehen!  
32 Bitte geben Sie ein kodierte Wort ein:  
33 76389g2a73  
34 Die Eingabe darf nur Ziffern von 0–9 enthalten!  
35 Es muss genau ein Leerzeichen (1) oder Punkt (0) am  
Wortende stehen!  
36 Bitte geben Sie ein kodierte Wort ein:  
37 76389g2a703  
38 Die Eingabe darf nur Ziffern von 0–9 enthalten!  
39 Nach dem Leerzeichen bzw. Punkt duerfen keine weiteren  
Zeichen folgen!  
40 Bitte geben Sie ein kodierte Wort ein:

```
41 76389g2a7310
42 Die Eingabe darf nur Ziffern von 0–9 enthalten!
43 Es muss genau ein Leerzeichen (1) oder Punkt (0) am
   Wortende stehen!
44 Bitte geben Sie ein kodierte Wort ein:
45 76389027310
46 Es muss genau ein Leerzeichen (1) oder Punkt (0) am
   Wortende stehen!
47 Bitte geben Sie ein kodierte Wort ein:
48 76389g2 a73
49 Die Eingabe darf nur Ziffern von 0–9 enthalten!
50 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
51 Es muss genau ein Leerzeichen (1) oder Punkt (0) am
   Wortende stehen!
52 Bitte geben Sie ein kodierte Wort ein:
53 763892730
54 Im Woerterbuch wurde keine passender Eintrag gefunden!
55 Bitte Wort im Explizit–Modus eingeben:
56 74633381912173320
57 SOFTWARE
58 Wort in Ordnung? ja (*) / nein (#)
59 *
60 Das Wort SOFTWARE wurde mit Code 76389273 im Woerterbuch
   abgespeichert!
61 Eingebener Satz:
62 SOFTWARE.
```

## Grenzfälle

**T2.16** Zu kurze Eingabe.

```
1 Bitte geben Sie ein kodierte Wort ein:
2 1
3 Das Wort sollte mindestens einen Buchstaben enthalten!
```

4 Bitte geben Sie ein kodierte Wort ein:

**T2.17** Eingabe ist zu lang.

1 Bitte geben Sie ein kodierte Wort ein:

2 763892732323332240

3 Wort zu lang! Es sind höchstens 16 Ziffern erlaubt (15 Buchstaben und 1 Wort-Ende)!

4 Es muss genau ein Leerzeichen (1) oder Punkt (0) am Wortende stehen!

5 Bitte geben Sie ein kodierte Wort ein:

## 4.3 Explizitmodus-Eingabe-Tests

In den folgenden Testfällen wird geprüft, ob das Programm bei Verletzung der in Kapitel 1.3.2 syntaktischen Regeln für den Explizitmodus, passend reagiert. In allen Fällen sollte ein Fehler gemeldet und eine neue Eingabe gefordert werden.

### Fehlerfälle

**T3.1** Unerlaubte Zeichen.

1 Bitte Wort im Explizit-Modus eingeben:

2 3343g7423420

3 Die Eingabe darf nur Ziffern von 0–9 enthalten!

4 Bitte Wort im Explizit-Modus eingeben:

**T3.2** Leerzeichen im Wort.

1 Bitte Wort im Explizit-Modus eingeben:

2 3343 7423420

3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!

4 Bitte Wort im Explizit-Modus eingeben:

**T3.3** Gerade Anzahl von Ziffern.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 334327423420
- 3 Es wurde eine gerade Zahl von Ziffern eingegeben. Im Explizitmodus werden buchstaben als Paare kodiert, das Abschliessende Symbol (Punkt (0) oder Leerzeichen (1)) wird aber nur einmal eingegeben!
- 4 Bitte Wort im Explizit-Modus eingeben:

**T3.4** Ungültiges Wortende.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 33437423423
- 3 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1) abgeschlossen!
- 4 Bitte Wort im Explizit-Modus eingeben:

**T3.5** Unerlaubte Zeichen und Leerzeichen im Wort.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 3343g74 23420
- 3 Die Eingabe darf nur Ziffern von 0-9 enthalten!
- 4 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 5 Bitte Wort im Explizit-Modus eingeben:

**T3.6** Unerlaubte Zeichen und gerade Anzahl von Ziffern.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 3343g743420
- 3 Die Eingabe darf nur Ziffern von 0-9 enthalten!
- 4 Es wurde eine gerade Zahl von Ziffern eingegeben. Im Explizitmodus werden buchstaben als Paare kodiert, das Abschliessende Symbol (Punkt (0) oder Leerzeichen (1)) wird aber nur einmal eingegeben!



5 Bitte Wort im Explizit-Modus eingeben:

**T3.7** Unerlaubte Zeichen und ungultiges Wortende.

1 Bitte Wort im Explizit-Modus eingeben:  
 2 3343g7423424  
 3 Die Eingabe darf nur Ziffern von 0-9 enthalten!  
 4 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
 abgeschlossen!  
 5 Bitte Wort im Explizit-Modus eingeben:

**T3.8** Leerzeichen im Wort und gerade Anzahl von Ziffern.

1 Bitte Wort im Explizit-Modus eingeben:  
 2 33432 7423420  
 3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
 4 Es wurde eine gerade Zahl von Ziffern eingeben. Im  
 Explizitmodus werden buchstaben als Paare kodiert, das  
 Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))  
 wird aber nur einmal eingegeben!  
 5 Bitte Wort im Explizit-Modus eingeben:

**T3.9** Leerzeichen im Wort und ungultiges Wortende.

1 Bitte Wort im Explizit-Modus eingeben:  
 2 334374 23423  
 3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
 4 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
 abgeschlossen!  
 5 Bitte Wort im Explizit-Modus eingeben:

**T3.10** Gerade Anzahl von Ziffern und ungultiges Wortende.

1 Bitte Wort im Explizit-Modus eingeben:  
 2 3343742342

- 3 Es wurde eine gerade Zahl von Ziffern eingegeben. Im Explizitmodus werden buchstaben als Paare kodiert, das Abschliessende Symbol (Punkt (0) oder Leerzeichen (1)) wird aber nur einmal eingegeben!
- 4 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1) abgeschlossen!
- 5 Bitte Wort im Explizit-Modus eingeben:

**T3.11** Unerlaubte Zeichen und Leerzeichen im Wort und gerade Anzahl von Ziffern.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 3343g74 234230
- 3 Die Eingabe darf nur Ziffern von 0–9 enthalten!
- 4 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 5 Es wurde eine gerade Zahl von Ziffern eingegeben. Im Explizitmodus werden buchstaben als Paare kodiert, das Abschliessende Symbol (Punkt (0) oder Leerzeichen (1)) wird aber nur einmal eingegeben!
- 6 Bitte Wort im Explizit-Modus eingeben:

**T3.12** Unerlaubte Zeichen und Leerzeichen im Wort und ungültiges Wortende.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 3343g74 23423
- 3 Die Eingabe darf nur Ziffern von 0–9 enthalten!
- 4 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 5 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1) abgeschlossen!
- 6 Bitte Wort im Explizit-Modus eingeben:

**T3.13** Gerade Anzahl von Ziffern und Leerzeichen im Wort und ungültiges Wortende.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 334374 2342
- 3 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 4 Es wurde eine gerade Zahl von Ziffern eingeben. Im  
Explizitmodus werden buchstaben als Paare kodiert, das  
Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))  
wird aber nur einmal eingegeben!
- 5 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
abgeschlossen!
- 6 Bitte Wort im Explizit-Modus eingeben:

**T3.14** Gerade Anzahl von Ziffern und Leerzeichen im Wort und ungültiges Wortende und ungültige Zeichen.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 3343z74 2342
- 3 Die Eingabe darf nur Ziffern von 0-9 enthalten!
- 4 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
- 5 Es wurde eine gerade Zahl von Ziffern eingeben. Im  
Explizitmodus werden buchstaben als Paare kodiert, das  
Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))  
wird aber nur einmal eingegeben!
- 6 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
abgeschlossen!
- 7 Bitte Wort im Explizit-Modus eingeben:

**T3.15** Alle oben stehenden Fehler hintereinander.

- 1 Bitte Wort im Explizit-Modus eingeben:
- 2 3343g7423420
- 3 Die Eingabe darf nur Ziffern von 0-9 enthalten!
- 4 Bitte Wort im Explizit-Modus eingeben:
- 5 3343 7423420
- 6 Leerzeichen innerhalb des Wortes sind nicht erlaubt!

7 Bitte Wort im Explizit-Modus eingeben:  
8 334327423420  
9 Es wurde eine gerade Zahl von Ziffern eingegeben. Im  
Explizitmodus werden buchstaben als Paare kodiert, das  
Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))  
wird aber nur einmal eingegeben!  
10 Bitte Wort im Explizit-Modus eingeben:  
11 33437423423  
12 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
abgeschlossen!  
13 Bitte Wort im Explizit-Modus eingeben:  
14 3343g74 23420  
15 Die Eingabe darf nur Ziffern von 0-9 enthalten!  
16 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
17 Bitte Wort im Explizit-Modus eingeben:  
18 3343g743420  
19 Die Eingabe darf nur Ziffern von 0-9 enthalten!  
20 Es wurde eine gerade Zahl von Ziffern eingegeben. Im  
Explizitmodus werden buchstaben als Paare kodiert, das  
Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))  
wird aber nur einmal eingegeben!  
21 Bitte Wort im Explizit-Modus eingeben:  
22 3343g7423424  
23 Die Eingabe darf nur Ziffern von 0-9 enthalten!  
24 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
abgeschlossen!  
25 Bitte Wort im Explizit-Modus eingeben:  
26 33432 7423420  
27 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
28 Es wurde eine gerade Zahl von Ziffern eingegeben. Im  
Explizitmodus werden buchstaben als Paare kodiert, das  
Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))  
wird aber nur einmal eingegeben!

29 Bitte Wort im Explizit-Modus eingeben:  
30 334374 23423  
31 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
32 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
abgeschlossen!  
33 Bitte Wort im Explizit-Modus eingeben:  
34 3343742342  
35 Es wurde eine gerade Zahl von Ziffern eingeben. Im  
Explizitmodus werden buchstaben als Paare kodiert, das  
Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))  
wird aber nur einmal eingegeben!  
36 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
abgeschlossen!  
37 Bitte Wort im Explizit-Modus eingeben:  
38 3343g74 234230  
39 Die Eingabe darf nur Ziffern von 0-9 enthalten!  
40 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
41 Es wurde eine gerade Zahl von Ziffern eingeben. Im  
Explizitmodus werden buchstaben als Paare kodiert, das  
Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))  
wird aber nur einmal eingegeben!  
42 Bitte Wort im Explizit-Modus eingeben:  
43 3343g74 23423  
44 Die Eingabe darf nur Ziffern von 0-9 enthalten!  
45 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
46 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)  
abgeschlossen!  
47 Bitte Wort im Explizit-Modus eingeben:  
48 334374 2342  
49 Leerzeichen innerhalb des Wortes sind nicht erlaubt!  
50 Es wurde eine gerade Zahl von Ziffern eingeben. Im  
Explizitmodus werden buchstaben als Paare kodiert, das  
Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))

```

    wird aber nur einmal eingegeben!
51 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)
    abgeschlossen!
52 Bitte Wort im Explizit-Modus eingeben:
53 3343z74 2342
54 Die Eingabe darf nur Ziffern von 0-9 enthalten!
55 Leerzeichen innerhalb des Wortes sind nicht erlaubt!
56 Es wurde eine gerade Zahl von Ziffern eingeben. Im
    Explizitmodus werden buchstaben als Paare kodiert, das
    Abschliessende Symbol (Punkt (0) oder Leerzeichen (1))
    wird aber nur einmal eingegeben!
57 Das Wort wurde mit keinem Gueltigen Zeichen (0 oder 1)
    abgeschlossen!
58 Bitte Wort im Explizit-Modus eingeben:
59 33437423420
60 FISCH
61 Wort in Ordnung? ja (*) / nein (#)
62 *
63 Das Wort FISCH wurde mit Code 34724 im Woerterbuch
    abgespeichert!
64 Eingegebener Satz:
65 FISCH.

```

## Grenzfälle

**T3.16** Grenzfall: Zu kurze Eingabe.

```

1 Bitte Wort im Explizit-Modus eingeben:
2 1
3 Das Wort sollte mindestens einen Buchstaben enthalten!
4 Bitte Wort im Explizit-Modus eingeben:

```

**T3.17** Grenzfall: Eingabe ist zu lang.



**T4.3** Mehrere ungültige Buchstaben.

```
1 Bitte Wort im Explizit-Modus eingeben :
2 64248142320
3 64 ist kein gueltiger Buchstabe!
4 24 ist kein gueltiger Buchstabe!
5 Bitte Wort im Explizit-Modus eingeben :
```

**T4.4** Ungültiger Buchstabe und Index größer 4.

```
1 Bitte Wort im Explizit-Modus eingeben :
2 65118142320
3 Buchstabenindex 5 des 01-ten Buchstaben zu hoch! Tasten
  haben maximal 4 Eintraege!
4 11 ist kein gueltiger Buchstabe!
5 Bitte Wort im Explizit-Modus eingeben :
```

**T4.5** Mehrere Indizes größer als 4.

```
1 Bitte Wort im Explizit-Modus eingeben :
2 65218142390
3 Buchstabenindex 5 des 01-ten Buchstaben zu hoch! Tasten
  haben maximal 4 Eintraege!
4 Buchstabenindex 9 des 05-ten Buchstaben zu hoch! Tasten
  haben maximal 4 Eintraege!
5 Bitte Wort im Explizit-Modus eingeben :
```

**Grenzfälle**

**T4.6** Wort wird nicht dem Wörterbuch hinzugefügt, da es bereits voll ist. Es wird trotzdem dem Satz angehängt.

```
1 Bitte Wort im Explizit-Modus eingeben :
2 61218142320
3 MATHE
```



```

4 Wort in Ordnung? ja (*) / nein (#)
5 *
6 Das Wort konnte nicht abgespeichert werden, weil das
  Woerterbuch voll ist!
7 Eingegebener Satz:
8 MATHE.

```

## 4.5 Wortvorschläge-Tests

### Normalfälle

Wörter, die einem eingegebenen T9-Code entsprechen, sollen in der Reihenfolge ihrer Häufigkeit ausgegeben werden.

**T5.1** Ausgabe der Wörter in richtiger Reihenfolge. Das Wörterbuch enthält mehrere Wörter zu einem T9-Code (Zur Übersicht wurden Wörter gewählt, die nicht zum T9-Code passen):

2345 DREI 3
2345 ZWEI 9
2345 VIER 2
2345 FUENF 1
2345 EINS 13

```

1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /
  nein (#)
2 *
3 Bitte geben sie den Dateinamen des Woerterbuchs ein.
4 TestWBs/WBuch-Reihenfolge.txt
5 Woerterbuch erfolgreich eingelesen.
6 Beginnen Sie einen neuen Satz oder beenden Sie das Programm
  mit (0).
7 Bitte geben Sie ein kodiertes Wort ein:
8 23450

```

```
9 EINS
10 Wort in Ordnung? ja (*) / nein (#)
11 #
12 ZWEI
13 Wort in Ordnung? ja (*) / nein (#)
14 #
15 DREI
16 Wort in Ordnung? ja (*) / nein (#)
17 #
18 VIER
19 Wort in Ordnung? ja (*) / nein (#)
20 #
21 FUENF
22 Wort in Ordnung? ja (*) / nein (#)
23 *
24 Eingebener Satz:
25 FUENF.
```

## 4.6 Wörterbücher-Tests

### Normalfälle

**T6.1** 6.1: Einlesen eines existierenden Wörterbuchs funktioniert fehlerfrei

```
1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /
  nein (#)
2 *
3 Bitte geben Sie den Dateinamen des Woerterbuchs ein.
4 TestWBs/WBuch-correct.txt
5 Woerterbuch erfolgreich eingelesen.
```

**T6.1** 6.1.1: Alle Einträge des Wörterbuchs sind verfügbar:

```
1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /  
   nein (#)  
2 *  
3 Bitte geben Sie den Dateinamen des Woerterbuchs ein.  
4 TestWBs/WBuch-correct.txt  
5 Woerterbuch erfolgreich eingelesen.  
6 Beginnen Sie einen neuen Satz oder beenden Sie das Programm  
   mit (0).  
7 Bitte geben Sie ein kodierte Wort ein:  
8 763892731  
9 SOFTWARE  
10 Wort in Ordnung? ja (*) / nein (#)  
11 *  
12 Bitte geben Sie ein kodierte Wort ein:  
13 96781  
14 WORT  
15 Wort in Ordnung? ja (*) / nein (#)  
16 *  
17 Bitte geben Sie ein kodierte Wort ein:  
18 3371  
19 DER  
20 Wort in Ordnung? ja (*) / nein (#)  
21 *  
22 Bitte geben Sie ein kodierte Wort ein:  
23 3370  
24 DER  
25 Wort in Ordnung? ja (*) / nein (#)  
26 #  
27 FES  
28 Wort in Ordnung? ja (*) / nein (#)  
29 *  
30 Eingebener Satz:  
31 SOFTWARE WORT DER FES.
```

## Fehlerfälle

**T6.2** Einlesen eines korrupten Wörterbuchs erzeugt Fehlermeldung.

```
1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /  
   nein (#)  
2 *  
3 Bitte geben Sie den Dateinamen des Woerterbuchs ein .  
4 TestWBs/WBuch-incorrect.txt  
5 Der 1-te Satz der Datei liegt nicht im richtigen Format vor  
   !  
6 Wenn mit einem beschaedigten Woerterbuch gearbeitet wird ,  
   kann es zur Laufzeit zu Fehlern kommen! Beheben Sie die  
   Syntaxfehler , lesen Sie ein anderes Woerterbuch ein oder  
   fahren Sie fort ohne Woerterbuch .
```

**T6.3** Einlesen eines nicht existierenden Wörterbuchs erzeugt Fehlermeldung.

```
1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /  
   nein (#)  
2 *  
3 Bitte geben Sie den Dateinamen des Woerterbuchs ein .  
4 IDoNotExist.txt  
5 Kein Woerterbuch geladen .
```

**T6.4** Einlesen eines leeren Wörterbuchs erzeugt Fehlermeldung.

```
1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /  
   nein (#)  
2 *  
3 Bitte geben Sie den Dateinamen des Woerterbuchs ein .  
4 TestWBs/WBuch-empty.txt  
5 Woerterbuch ist leer .  
6 Woerterbuch erfolgreich eingelesen .
```

## Sonderfälle

**T6.5** Nach Programmende enthält Wörterbuch sowohl eingelesene als auch neue Einträge.

```
1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /  
   nein (#)  
2 *  
3 Bitte geben Sie den Dateinamen des Woerterbuchs ein.  
4 TestWBs/WBuch-correct.txt  
5 Woerterbuch erfolgreich eingelesen.  
6 Beginnen Sie einen neuen Satz oder beenden Sie das Programm  
   mit (0).  
7 Bitte geben Sie ein kodierte Wort ein:  
8 2581  
9 ALT  
10 Wort in Ordnung? ja (*) / nein (#)  
11 *  
12 Bitte geben Sie ein kodierte Wort ein:  
13 4241  
14 Im Woerterbuch wurde kein passender Eintrag gefunden!  
15 Bitte Wort im Explizit-Modus eingeben:  
16 4323421  
17 ICH  
18 Wort in Ordnung? ja (*) / nein (#)  
19 *  
20 Das Wort ICH wurde mit Code 424 im Woerterbuch  
   abgespeichert!  
21 Bitte geben Sie ein kodierte Wort ein:  
22 2461  
23 Im Woerterbuch wurde kein passender Eintrag gefunden!  
24 Bitte Wort im Explizit-Modus eingeben:  
25 2243621  
26 BIN
```

```

27 Wort in Ordnung? ja (*) / nein (#)
28 *
29 Das Wort BIN wurde mit Code 246 im Woerterbuch
    abgespeichert!
30 Bitte geben Sie ein kodierte Wort ein:
31 6380
32 Im Woerterbuch wurde kein passender Eintrag gefunden!
33 Bitte Wort im Explizit-Modus eingeben:
34 6232820
35 NEU
36 Wort in Ordnung? ja (*) / nein (#)
37 *
38 Das Wort NEU wurde mit Code 638 im Woerterbuch
    abgespeichert!
39 Eingegebener Satz:
40 ALT ICH BIN NEU.

```

258 ALT 1
246 BIN 1
424 ICH 1
638 NEU 1

Abbildung 4.3: Wörterbuch nach Test 6.5

## Grenzfall

**T6.2** Grenzfall: Textdatei hat mehr Einträge als interne Datenstruktur. Erwartet wird eine Meldung, es wird aber keine erneute Eingabe gefordert.

```

1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /
    nein (#)
2 *
3 Bitte geben Sie den Dateinamen des Woerterbuchs ein.
4 TestWBs/WBuch-401-Eintraege.txt

```

- ```
5 Alle Eintraege ab dem 401-ten Satz der Datei wurden nicht
   eingelesen , da das Woerterbuch voll ist .
6 Woerterbuch erfolgreich eingelesen .
```

## 4.7 Sonderfälle

Nun werden die bereits beschriebenen Sonderfälle (Kapitel 1.4.4) getestet.

### Normalfälle

**T7.1** Der Anwender gibt im Explizitmodus etwas ein, was nicht der initialen Eingabe im normalen Modus entspricht. Erwartet wird eine Meldung, die darauf hinweist und das Speichern des Wortes aus der Explizit-Eingabe.

- ```
1 Bitte geben Sie ein kodierte Wort ein :
2 3370
3 Im Woerterbuch wurde kein passender Eintrag gefunden !
4 Bitte Wort im Explizit-Modus eingeben :
5 325332332162810
6 ELEFANT
7 Wort in Ordnung? ja (*) / nein (#)
8 *
9 Das explizit eingegebene Wort entspricht nicht der
   originalen Eingabe .
10 Das Wort ELEFANT wurde mit Code 3533268 im Woerterbuch
   abgespeichert !
11 Eingegebener Satz :
12 ELEFANT .
```

**T7.2** Der Anwender gibt im Explizitmodus etwas ein, was nicht der initialen Eingabe im normalen Modus entspricht. Erwartet wird eine Meldung, die darauf hinweist. Das Explizit-Wort ist bereits im Wörterbuch vorhanden. Deshalb wird nur die Häufigkeit erhöht.

3533268 ELEFANT 1

**Abbildung 4.4:** Wörterbuch vor Test 7.2.

```
1 Moechten Sie ein externes Woerterbuch einlesen? ja (*) /  
   nein (#)  
2 *  
3 Bitte geben Sie den Dateinamen des Woerterbuchs ein.  
4 TestWBs/WBuch-Sonderfall.txt  
5 Woerterbuch erfolgreich eingelesen.  
6 Beginnen Sie einen neuen Satz oder beenden Sie das Programm  
   mit (0).  
7 Bitte geben Sie ein kodierte Wort ein:  
8 3370  
9 Im Woerterbuch wurde kein passender Eintrag gefunden!  
10 Bitte Wort im Explizit-Modus eingeben:  
11 325332332162810  
12 ELEFANT  
13 Wort in Ordnung? ja (*) / nein (#)  
14 *  
15 Das explizit eingegebene Wort entspricht nicht der  
   originalen Eingabe.  
16 Wort war bereits im Woerterbuch vorhanden.  
17 Eingegebener Satz:  
18 ELEFANT.  
19 Beginnen Sie einen neuen Satz oder beenden Sie das Programm  
   mit (0).  
20 Bitte geben Sie ein kodierte Wort ein:  
21 0  
22 Programm Ende.
```

3533268 ELEFANT 2

**Abbildung 4.5:** Wörterbuch nach Test 7.2.



# 5 Zusammenfassung und Ausblick

## 5.1 Zusammenfassung

Es wurde eine Software angefertigt, welche alle Vorgaben bezüglich der geforderten Funktionalität erfüllt. Der gesamte Ablauf, vom Einlesen eines externen Wörterbuchs, über das eigentliche Mapping der Tasten und der Datenstruktur, bis hin zur Behandlung von Fehlerfällen und das Schreiben eines neuen Wörterbuchs, wurde korrekt abgebildet.

Mithilfe dieser Entwicklung können vereinfacht Texteingaben auf Tastaturen mit gleichem Aufbau, wie die eines Mobiltelefons, getätigt werden.

Bei der Implementierung wurde stets darauf geachtet, die Modularisierung einzuhalten, um mögliche Erweiterungen einfach einbinden zu können. Eine ausführliche Test- und Entwicklungsdokumentation und ein Programmablaufplan geben projektfremden Personen, besonders Entwicklern einen tiefen Einblick in die Software und ihrer Funktionsweise. Weitergehend wurden sprechende Variablennamen mit Namenskonventionen gewählt, um die Wartbarkeit des Codes maximal zu halten.

## 5.2 Ausblick

Die entwickelte Software kann vielseitig erweitert und verbessert werden.

Eine mögliche Verbesserung wäre eine striktere Kontrolle der externen Wörterbücher. Aktuell werden die Dateien nur minimal auf syntaktische Anforderungen überprüft, der Inhalt der einzelnen Zeilen wird aber nicht validiert.

Des Weiteren könnte die Größe des internen Wörterbuchs dynamische zur Laufzeit erweitert werden. Im Moment gibt es eine feste maximale Größe, sodass das

Wörterbuch irgendwann voll ist. Die Tabellenstrukturen liegen dafür schon in der passenden Form vor.

Auch Programmausgaben im Benutzerdialog haben Verbesserungspotential. Alle eingegebenen Sätze könnten zum Beispiel als Paragraf am Programmende ausgegeben werden.

Zudem könnte ein noch größerer Fokus auf COBOL-Code-Konventionen gelegt werden.

# A Benutzeranleitung

## A.1 Vorbereiten des Systems

### A.1.1 Systemvoraussetzungen

Um das Programm zu benutzen ist ein Windows- oder Linux-System vorausgesetzt. Unter Linux muss zusätzlich *Wine* installiert werden.

### A.1.2 Installation

Die Installation des Programms erfolgt über das Entpacken der .zip-Datei. Die ausführbare .exe befindet sich im Anschluss im Unterordner *bin*. Außerdem ist es notwendig, dass unter Windows die PATH-Variable den Pfad zu GnuCOBOL/bin enthält.

## A.2 Programmaufruf

Um das Programm zu starten, muss die bin/TexteingabeHandy.exe aufgerufen werden. Es öffnet sich ein Dialogfenster, mit welchem der Nutzer anschließend interagieren kann.

## A.3 Testen der Beispiele

Die Beispiele können alle mittels der .cmd-Dateien im bin Verzeichnis getestet werden. Die TestAll.cmd enthält dabei alle Testfälle, welche automatisch nacheinander ausgeführt werden. Hilfreich ist es, die ScreenBuffer-Size der Eingabeaufforderung

auf eine höhere Zahl zu setzen, damit alle Zeilen in dem Dialogfenster bestehen bleiben. Zu empfehlen ist hierbei eine Zahl über 1000.

## B Entwicklungsumgebung

Das Programm wurde mithilfe der OpenCobolIDE (<https://launchpad.net/cobcide/+download>) in der Version 4.7.6 geschrieben. Dabei handelt es sich um eine leichtgewichtige COBOL Entwicklungsumgebung, die als Compiler GnuCOBOL 2.0.0 (<https://sourceforge.net/projects/gnucobol/>) verwendet. Der GnuCOBOL-Compiler übersetzt den COBOL-Source-Code in ein C-Programm und erzeugt daraus, mit einem nativen C-Compiler, eine ausführbare Datei.

Im Entwicklungsprozess wurde zur Versionsverwaltung GitHub (<https://github.com/>) verwendet. Die dort angebotenen Remote-Repositories ermöglichen eine effiziente Zusammenarbeit im Team.

Alle Entwicklungsschritte wurden auf Systemen mit Windows 10 Betriebssystem (<https://www.microsoft.com/de-de/software-download/windows10>) durchgeführt.



**Abbildung B.1:** Logos von OpenCobolIDE und GnuCOBOL.



**Abbildung B.2:** GitHub-Logo.

## C Verwendete Hilfsmittel

Als Hilfsmittel wurden hauptsächlich die Inhalte der, von Prof. Dr. rer. nat. Karola Merkel (<https://www.fh-aachen.de/fachbereiche/medizintechnik-und-technomathematik/einrichtungen/sp-studienort-koeln/kontakt>) angeboten, Vorlesung „COBOL“ verwendet. Ergänzend dazu wurde die offizielle COBOL-Dokumentation von IBM (<https://www.ibm.com/docs/en>) zurate gezogen.

Zudem konnten unterschiedliche Fragen durch das Durchsuchen von Foren gelöst werden. Besonders häufig konnten das „Expertforum“ (<https://ibmmainframes.com/forum-1.html>) und „stackoverflow“ ([stackoverflow.com](https://stackoverflow.com)) Antworten liefern.

# D Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Thema

*COBOL*

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, sind kenntlich gemacht und die Arbeit ist in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung.

Mein Beitrag zur Abgabe:

- Ich habe große Teile des Benutzerdialogs implementiert.  
COBOL-Paragrafen: „Benutzer-Dialog“, „Ermittle-Wort“, „Wort-Auswahl“, „Finde-Moeglichkeiten“, „Sortiere-Nach-Haeuf“, „Explizit-Eingabe“, „Suche-Wort-In-WBuch“ und „Konstruiere-Wort“
- Kapitel der Dokumentation: 1.3, 1.4, 2.1.4, 4.1, 4.3, 4.4, 4.5, 4.7, B, C
- automatische Tests: TestExplizitEingabe.cmd, TestFunktionalitaet.cmd, TestReihenfolgeWortvorschlag.cmd, TestSemantikExplizitEingabe.cmd und TestSonderfall.cmd

Köln, den 2. Dezember 2023

---

Ben Pietsch

Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Thema

*COBOL*

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, sind kenntlich gemacht und die Arbeit ist in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung.

Mein Beitrag zur Abgabe:

- Validierung der Benutzereingaben:  
COBOL-Paragrafen: „Pruefe-Eingabe“ und „Pruefe-Explizit-Eingabe“
- Kapitel der Dokumentation: 3, 5,
- automatische Tests: TestT9-Eingabe.cmd

Köln, den 2. Dezember 2023

---

Natalie Fritzen



Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Thema

*COBOL*

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, sind kenntlich gemacht und die Arbeit ist in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung.

Mein Beitrag zur Abgabe:

- Einlesen von externen Wörterbüchern und das sortieren und schreiben des internen Wörterbuchs.  
COBOL-Paragrafen: „Auswahl-WBuch“, „Einlesen-WBuch.“, „Lese-Satz“ und „Schreibe-WBuch-Sortiert“
- Kapitel der Dokumentation: 1.1, 1.2, 2.1.1, 2.1.2, 2.1.3, 4.6, A
- automatische Tests: TestAlle.cmd, und TestWBuchEinlesen.cmd

Köln, den 2. Dezember 2023

---

Leonhard Keßler

## **E Aufgabenstellung**

## **F Quellcode**