



Fachhochschule Aachen

Studienort Köln

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

Projektaufgabe COBOL

Gruppenwechsel

Projektarbeit

von

Leon Jarosch

Matrikelnummer: 3283258

Sankt Augustin, den 19. Dezember 2023

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Aufgabenanalyse | 4 |
| 1.1 | Interpretation der Aufgabe | 4 |
| 1.2 | Anforderung an das Programm | 5 |
| 1.3 | Fehlerarten | 6 |
| 2 | Verfahrensbeschreibung | 7 |
| 2.1 | Gesamtsystem | 7 |
| 2.1.1 | Eingabe | 7 |
| 2.1.2 | Verarbeitung | 7 |
| 2.2 | Datenstrukturen | 10 |
| 2.2.1 | Datenspeicherstrukturen | 10 |
| 2.2.2 | Ausgabestrukturen | 10 |
| 3 | Programmbeschreibung | 12 |
| 3.1 | Programmablaufplan | 12 |
| 3.2 | Entwicklungsdokumentation | 12 |
| 4 | Testdokumentation | 17 |
| 4.1 | Definierte Tests | 17 |
| 4.2 | Normalmodus-Tests | 23 |
| 4.3 | Explizitmodus-Eingabe-Tests | 29 |
| 4.4 | Semantik Explizitmodus-Tests | 37 |
| 4.5 | Wortvorschläge-Tests | 39 |
| 4.6 | Wörterbücher-Tests | 40 |
| 4.7 | Sonderfälle | 45 |

| | |
|---------------------------------------|-----------|
| A Benutzeranleitung | 47 |
| A.1 Vorbereiten des Systems | 47 |
| A.1.1 Systemvoraussetzungen | 47 |
| A.1.2 Installation | 47 |
| A.2 Programmaufruf | 47 |
| A.3 Testen der Beispiele | 47 |
| B Entwicklungsumgebung | 48 |
| C Verwendete Hilfsmittel | 49 |
| D Erklärung | 50 |
| E Aufgabenstellung | 51 |

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 zeilenweise Abrechnungen auflistet. Teil einer Abrechnung sind:

| | | | | | | | | | | |
|--|-------|--|----------|--|-------------|--|-------------|--|--------|--|
| | Datum | | KundenID | | LeistungsID | | Einzelpreis | | Anzahl | |
|--|-------|--|----------|--|-------------|--|-------------|--|--------|--|

Tabelle 1.1: Struktureller Aufbau der Tastatur

Die KundenID besteht dabei aus einem führenden „K“ folgend von einer fünfstelligen Nummer.

Die LeistungsID besteht aus sechs Ziffern.

Das Datum ist im Format „JJJJ.MM.TT“ angegeben, spielt für die Rechnung jedoch keine relevante Rolle.

Der Einzelpreis besitzt immer zwei Nachkommastellen und ist in Euro angegeben.

Die Anzahl ist eine ganze Zahl bis maximal 99.

Das einzulesende Journal ist bereits nach Kunden-ID und Leistungs-ID vorsortiert.

Für jeden erkannten Kunden soll nun ausgewertet werden, welche Leistungen in Anspruch genommen wurden und wie viel diese gekostet haben.

Abschließend soll eine Rechnung erstellt werden, welche die Gesamtkosten eines Kunden zusammenfasst. Dabei wird die Rechnung für den Kunden mit der Kunden-ID gekennzeichnet. Folgend wird eine Tabellenstruktur ausgegeben, welche die

einzelnen Leistungen und deren Kosten und weitere Informationen zur Leistung auflistet. Eine Leistungszeile besteht dabei aus folgenden Spalten:

| | | | | | | |
|----------|-------------|--------------------------|--------|-------------|-------------|--|
| Position | LeistungsID | Bezeichnung der Leistung | Anzahl | Einzelpreis | Gesamtpreis | |
|----------|-------------|--------------------------|--------|-------------|-------------|--|

Tabelle 1.2: Abrechnungszeile

Die Position ist eine Inkrementierung der Leistungen für einen Kunden. Leistungs-ID, Anzahl und Einzelpreis werden direkt aus dem Journal übernommen. Der Gesamtpreis ergibt sich aus der Multiplikation von Anzahl und Einzelpreis. Die Bezeichnungen der Leistungen sind in einer externen Datei abgelegt. Dieses Glossar beinhaltet zu allen bekannten Leistungs-IDs eine passende Bezeichnung

Abschließend werden aus allen erhobenen Leistungen und deren Gesamtpreis die Gesamtkosten des Kunden berechnet und ausgegeben.

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

1.2 Anforderung an das Programm

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

Es muss

- Ein Journal zeilenweise einlesen
- Die genutzten Leistungen für jeden Kunden ermitteln
- Aus den Leistungen die Gesamtkosten für jeden Kunden ermitteln
- Für jeden Kunden eine Rechnung erstellen
- Die Rechnungen in einer Rechnungsdatei speichern

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

1.3 Fehlerarten

2 Verfahrensbeschreibung

2.1 Gesamtsystem

Das System arbeitet in grundzügen 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. In diesem Fall gibt es jedoch keine explizite Trennung der drei Fälle, da die Phasen zeitgleich ablaufen. So wird die Eingabe Zeilenweise vollzogen und jede eingelesene Zeile sofort verarbeitet. Nach der verarbeitung wird falls möglich eine Ausgabezeile erstellt und ausgegeben. Jeder Schritt besteht also aus allen drei Phasen.

2.1.1 Eingabe

Die Journal Datei wird zeilenweise eingelesen. Dabei werden die Überschriften ignoriert und beginnend mit der ersten inhaltlichen Zeile die Verarbeitung gestartet.

2.1.2 Verarbeitung

Die Verarbeitung nutzt das Prinzip des Gruppenwechsels. Dieses lässt sich anwenden, da die einzulesende Datei bereits nach Kunden-ID und Leistungs-ID vorsortiert ist. Dabei wird die Datei zeilenweise eingelesen und etappenweise ausgewertet.

Initialisieren des Kunden

Im ersten Schritt wird der aktuelle Kunde ermittelt und initialisiert. Neben dem speichern der Kunden-ID, werden die Position und die Gesamtkosten in ihren

Startzustand versetzt. Beide Werten werden später genutzt.

Außerdem die Rechnungsdatei befüllt.

Dabei wird die Kunden-ID als Überschrift gesetzt und die Kopfzeile der Tabelle zum auflisten der jeweiligen Leistungen gedruckt.

Initialisieren der Leistung

Anschließend wird die aktuelle Leistung initialisiert. Analog zur Kunden Initialisierung wird die Leistungs-ID als Schlüssel gesetzt. Außerdem wird die zu sammelnde Anzahl, wie auch der aktuelle Gesamtpreis der Leistung in den Startzustand versetzt. Mit diesen wird ebenfalls erst später weitergearbeitet.

Nun wird die Position der Leistung um eins erhöht. Dieser Wert spiegelt die Anzahl an unterschiedlichen Leistungen eines Kunden wieder und wird mit jeder neuen Leistung inkrementiert.

Für die Ermittlung der Leistungsbezeichnung wird ein Unterprogramm aufgerufen. Siehe 2.1.2.

Verarbeitung

Im tiefsten Schritt des Gruppenwechsels wird die eingelesene Zeile final ausgewertet. Dazu wird die Anzahl der Leistung, um die im Satz angegebene Anzahl erhöht.

Die Datei wird nun zeilenweise weiter eingelesen.

Solange sich die beiden IDs nicht ändern, wird für jede Zeile die Anzahl der aktuellen Leistung hochgezählt. Anhand dieser kann dann später der Gesamtpreis der jeweiligen Leistung berechnet werden.

Dies geschieht solange, bis sich entweder die Kunden-ID oder die Leistungs-ID ändert.

Leistungs-ID ändert sich

Ändert sich die Leistungs-ID, so gilt die aktuelle Leistung als abgeschlossen. Dabei kann aus der ermittelten Anzahl und des gespeicherten Einzelpreises der Gesamtpreis der Leistung errechnet werden.

Damit sind alle Informationen einer Leistung bekannt und können in die Rechnung geschrieben werden.

Zum Abschluss wird der errechnete Gesamtpreis der Leistung auf die Gesamtkosten des Kunden addiert.

Nun wird der Prozess bei der Leistungsinitialisierung 2.1.2 mit der neuen Leistungs-ID fortgesetzt.

Kunden-ID ändert sich

Ändert sich die Kunden-ID, so gilt der aktuelle Kunde als abgeschlossen.

Aus der errechneten Gesamtkosten wird die Rechnung für den Kunden abgeschlossen und innerhalb der Rechnungsdatei abgetrennt.

Der Prozess wird nun in der Kundeninitialisierung 2.1.2 mit der neuen Kunden-ID fortgesetzt.

Sind alle Zeilen der Datei eingelesen, terminiert das Programm und schließt den Prozess ab.

Ermittlung der Leistungsbezeichnung

Initial wird die Leistungsbezeichnung mit „Unbekannt“ befüllt. Dies ist ein sicherheitsmechanismus, da nicht gewährleistet ist, dass jede Leistungs-ID dem Glossar bekannt ist. Da alle für die Rechnung relevanten Informationen bereits im Journal aufzufinden sind, soll es in diesem Fall keinen Programmabbruch geben

Um die Leistungsbezeichnung zu ermitteln, wird die Leistungs-ID mit dem Leistungsglossar verglichen.

Dazu wird das gennante Glossar zeilenweise eingelesen. Eine Zeile besteht dabei aus einer Leistungs-ID und einer mit „:“ getrennten Leistungsbeschreibung.

Wird im Glossar die gesuchte Leistungs-ID gefunden, so wird die Leistungsbezeichnung mit der zur ID passenden Leistungsbezeichnung überschrieben und zurück an den Hauptablauf übergeben.

2.2 Datenstrukturen

Die genutzten Datenstrukturen lassen sich in zwei Kategorien einteilen.

2.2.1 Datenspeicherstrukturen

Die Datenspeicherstrukturen dienen zum speichern und verarbeiten der Daten.

Diese lassen sich wiederum in zwei Kategorien einteilen.

Eingabespeicher

Im Zuge des Programms werden zwei Dateien eingelesen.

Für beide Dateien gibt es eine Datenstruktur die alle Informationen eines Datensatzes speichert.

Gedächtnisspeicher

Um den Gruppenwechsel wie beschrieben zu realisieren müssen die aktuellen Daten einer Rechnung gespeichert werden. Dies unterscheidet sich in die allgemeinen Rechnungsdaten und die Rechnungsdaten einer Leistung.

Die allgemeinen Rechnungsdaten dienen als Speicher für Kunden-ID und Gesamtsumme.

Die zeilenweisen Rechnungsdaten speichern die Leistungsanzahl wie auch die restlichen Daten einer Leistung.

2.2.2 Ausgabestrukturen

Für eine optisch angebrachte Ausgabe werden die Datenstrukturen in eine Ausgabeform gebracht.

Dieses überführt die Datenspeicher in eine optisch einsehbaaren Form. Außerdem werden die optischen Bausteine der Rechnung, wie die Tabellenkopfzeile, definiert.

3 Programmbeschreibung

3.1 Programmablaufplan

Die folgenden Abbildungen beschreiben Teile des Programms.

Abbildung 3.1 und 3.2 zeigen den Ablauf des Gruppenwechsels.

In Abbildung 3.3 wird die Ermittlung der Leistungsbezeichnungen visualisiert.

3.2 Entwicklungsdokumentation

Es wurden grundsätzlich sprechende Namen für Variablen, Abschnitte und Paragraphen gewählt. Außerdem sind `DISPLAY` Statements welche zum Debuggen genutzt wurden erhalten geblieben. Mit diesen ist der Programmablauf leichter nachzuvollziehen.

Die Funktionen der einzelnen Paragraphen sind in Tabelle 3.1 beschrieben.

| Bezeichnung | Beschreibung |
|-----------------------|---|
| MAIN-PROCEDURE | Hauptablauf welcher den Gruppenwechsel delegiert |
| PREPERATION | Spiegelt den Vorlauf zum einlesen einer Datei wieder und öffnet die Eingabe-(JOURNAL.txt) und Ausgabedatei (INVOICE.txt) |
| CUSTOMER-PREPERATION | Wertet den akutellen Einabesatz aus und initialisiert daraus einen Kunden |
| SERVICE-PREPERATION | Wertet den aktuellen Eingabesatz aus und initialisiert daraus eine Leistung |
| INDIVIDUAL-PROCESSING | Wertet den aktuellen Eingabesatz aus und zählt die Anzahl einer Leistung hoch |
| READ-NEXT-LINE | Liest, wenn möglich, die nächste Zeile des Journrals ein |
| SERVICE-COMPLETION | Wertet die aktuelle Leistung aus indem der Gesamtbetrag der Leistung berechnet und mit allen relevanten Leistungsdaten in die Rechnung geschrieben wird |
| CUSTOMER-COMPLETION | Wertet den aktuellen Kunden aus indem der Gesamtbetrag des Kunden in die Rechnung geschrieben und ein Ende gekennzeichnet wird |
| COMPLETION | schließt Einlese- und Auslesedatei |
| GET-SERVICE-TERM | Ermittelt anhand der Leistungs-ID durch einlesen des Leistungsglossar die Leistungsbeschreibung |
| CHECK-LINE | Überprüft, ob die aktuell aus dem Glossar eingelesene Leistungs-ID mit der zu überprüfenden Leistungs-ID übereinstimmt |
| NEXT-LINE | Liest, wenn möglich die nächste Zeile des Glossars ein |
| DISPLAY-JOURNAL | Debugereinheit zum ausgeben der aktuell eingelesenen Daten |

Tabelle 3.1: Aufgaben der einzelnen logischen Einheiten.

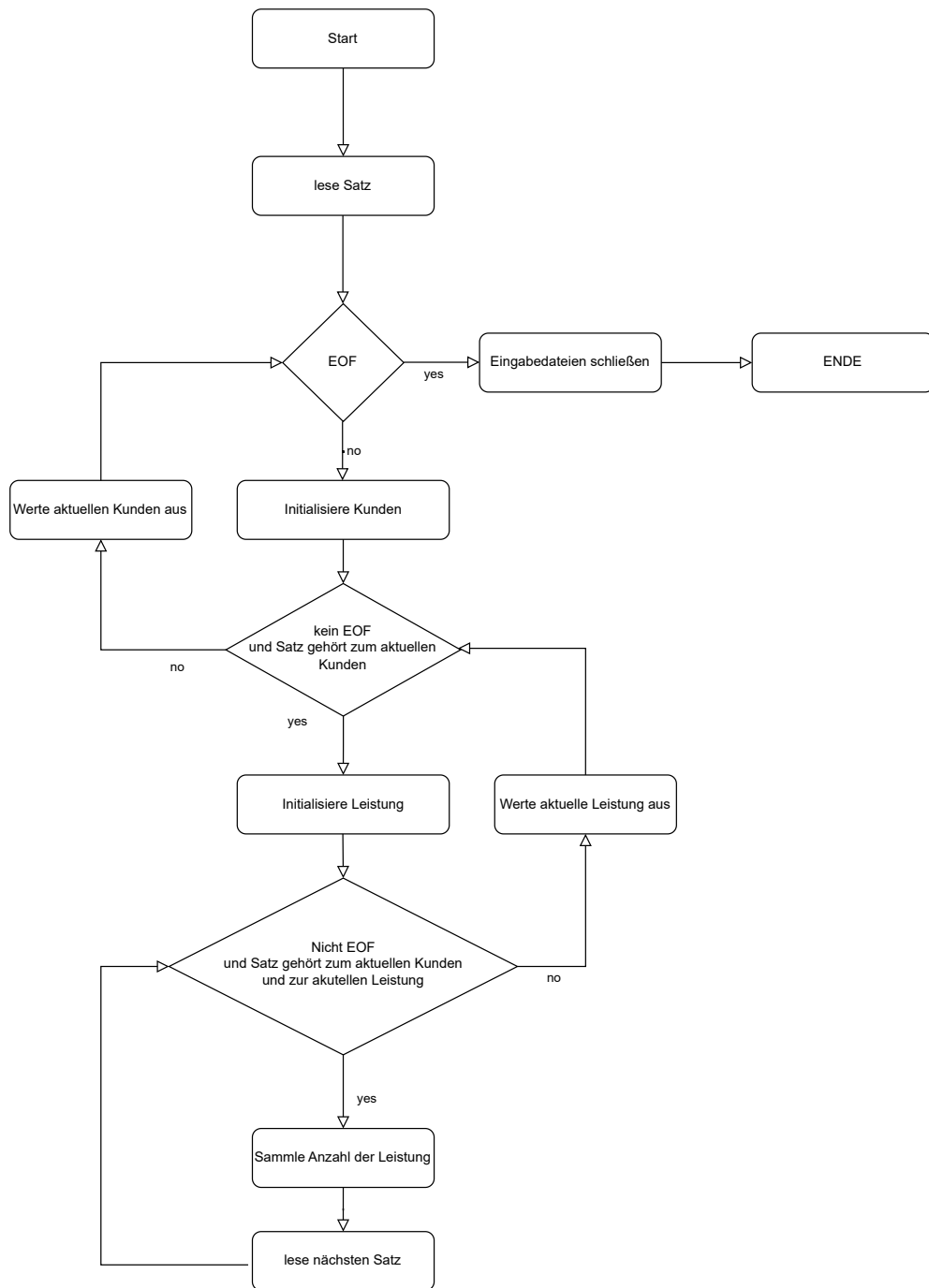


Abbildung 3.1: Veranschaulichung des Gruppenwechsels als Programmablaufplan

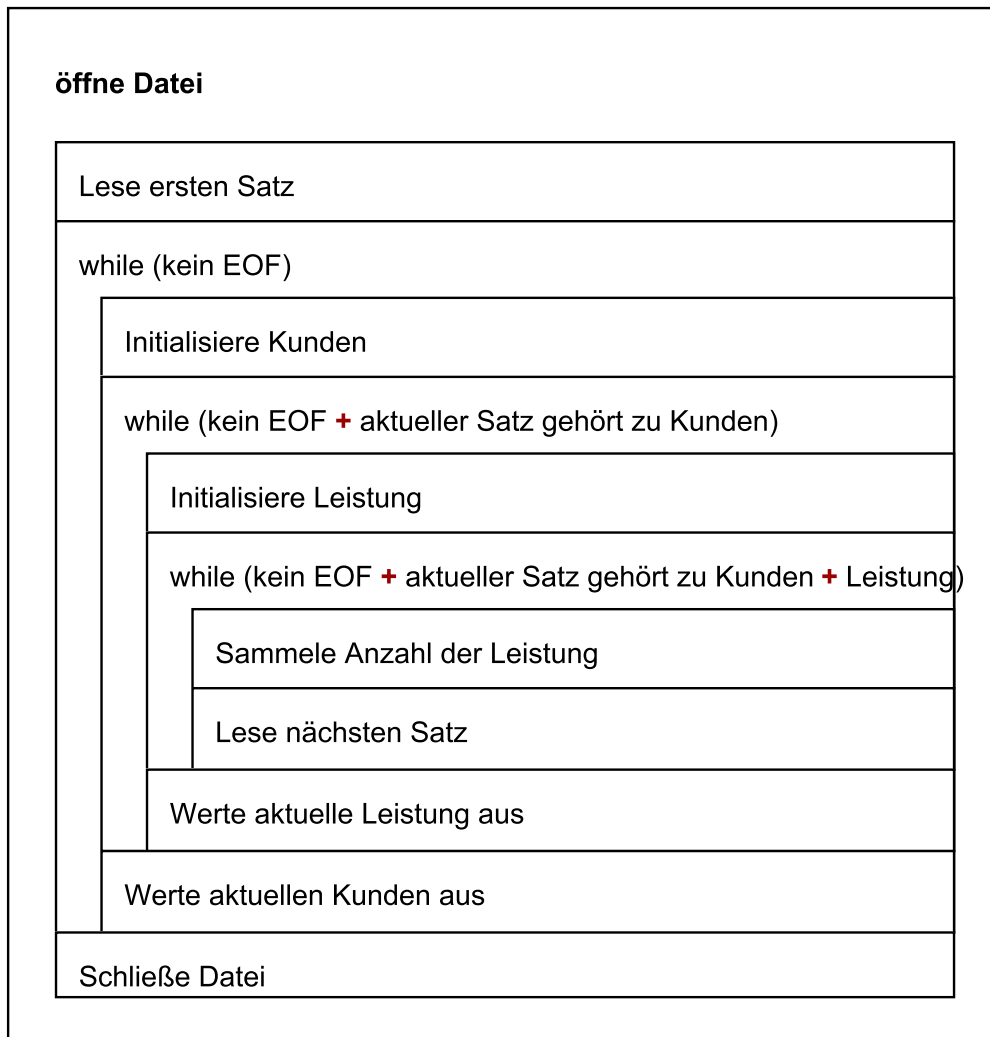


Abbildung 3.2: Veranschaulichung des Gruppenwechsels als Nassi-Schneiderman-Diagramm

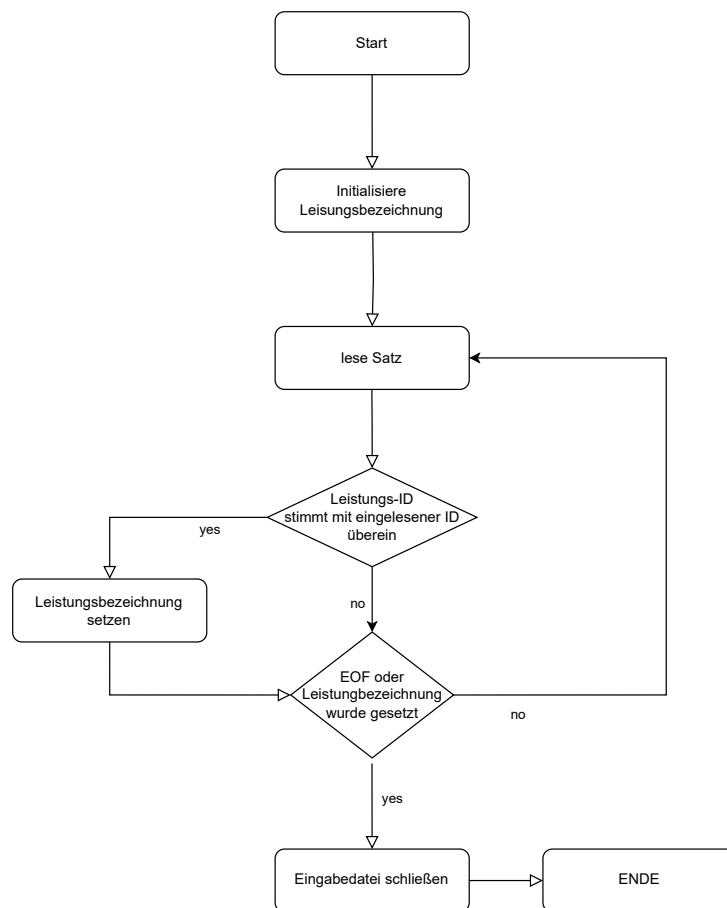


Abbildung 3.3: Ablauf der Ermittlung der Leistungsbezeichnungen

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 ??)

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 ?? 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 kodiertes Wort ein:
45 76389027310
46 Es muss genau ein Leerzeichen (1) oder Punkt (0) am
   Wortende stehen!
47 Bitte geben Sie ein kodiertes 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 kodiertes 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 kodiertes 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 ?? 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.

```

1 Bitte Wort im Explizit-Modus eingeben:
2 334374234222222222222222222233333333330
3 Wort zu lang! Es sind hoechstens 31 Ziffern erlaubt (15
  Buchstabenpaare und 1 Wort-Ende)!
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:

```

4.4 Semantik Explizitmodus-Tests

In den folgenden Tests werden Explizit-Eingaben geprüft, die den syntaktischen Vorgaben genügen, aber nicht dem strukturellen Aufbau der Handytastatur entsprechen.

Fehlerfälle

T4.1 Zugriff auf nicht existierenden Buchstaben 64.

```

1 Bitte Wort im Explizit-Modus eingeben:
2 64218142320
3 64 ist kein gueltiger Buchstabe!
4 Bitte Wort im Explizit-Modus eingeben:

```

T4.2 Index 65 außerhalb der maximalen Tastenlänge.

```

1 Bitte Wort im Explizit-Modus eingeben:
2 65218142320
3 Buchstabenindex 5 des 01-ten Buchstaben zu hoch! Tasten
   haben maximal 4 Eintraege!
4 Bitte Wort im Explizit-Modus eingeben:

```

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 Eingegebener 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 ??) 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.

A Benutzeranleitung

A.1 Vorbereiten des Systems

A.1.1 Systemvoraussetzungen

Um das Programm zu benutzen ist ein Windows-System vorausgesetzt.

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/Gruppenwechsel.exe aufgerufen werden.

A.3 Testen der Beispiele

Die Beispiele sind im Ordner *files/examples* zu finden. Um diese zu testen, müssen diese händisch in den Ordner *files* abgelegt und in *JOURNAL.txt* umbenannt werden. Anschließend kann das Programm gestartet werden. Alternativ ist zu empfehlen, die eigentliche *JOURNAL.txt* mit den Inhalt des zu testenden Beispiel zu überschreiben.

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.

Im Entwicklungsprozess wurde zur Versionsverwaltung GitHub (<https://github.com/>) verwendet. Die dort angebotenen Remote-Repositories ermöglichen eine Versionierung und Backups des Quellcodes.

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. Zudem konnten unterschiedliche Fragen durch das Durchsuchen von Foren gelöst werden. Besonders häufig konnte das „Expertforum „stackoverflow“ (stackoverflow.com) Antworten liefern.

D Erklärung

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

Projektaufgabe COBOL Gruppenwechsel

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.

Sankt Augustin, den 19. Dezember 2023

Leon Jarosch

E Aufgabenstellung

Projektarbeit 2023 „Gruppenwechsel“

Die Systemhaus GmbH soll ein Programm entwickeln, mit dem die Sportsfit AG Rechnungen für Vereinsmitglieder erstellen kann. Die abzurechnenden Leistungen sind in einem Journal gespeichert.

Journal (Beispiel)

| Datum | KundenID | LeistungsID | EinzelPreis | Anzahl |
|------------|----------|-------------|-------------|--------|
| 01.04.2016 | K00091 | 100076 | 2,40 | 2 |
| 10.04.2016 | K00091 | 100076 | 2,40 | 3 |
| 10.04.2016 | K00091 | 500123 | 15,00 | 1 |
| 03.04.2016 | K01234 | 200234 | 20,00 | 1 |
| 11.04.2016 | K01234 | 200234 | 20,00 | 1 |
| 05.04.2016 | K01234 | 200356 | 15,00 | 1 |

Das Journal ist nach KundenID und bei gleicher KundenID nach LeistungsID sortiert.

Die Rechnungen sollen wie folgt aufgebaut sein:

| K00091 | | | | | |
|------------------------|-------------|------------------------------|--------|-------------|-------------|
| Pos | LeistungsID | Bezeichnung der Leistung | Anzahl | Einzelpreis | Gesamtpreis |
| 1 | 100076 | Softgetränke | 5 | 2,40 | 12,00 |
| 2 | 500123 | Buch, „Fit ohne Anstrengung“ | 1 | 15,00 | 15,00 |
| Rechnungssumme (netto) | | | | | 27,00 |

| K01234 | | | | | |
|-------------------------|-------------|--------------------------|--------|-------------|-------------|
| Pos | LeistungsID | Bezeichnung der Leistung | Anzahl | Einzelpreis | Gesamtpreis |
| 1 | 200234 | Kraftraumstunde | 2 | 20,00 | 40,00 |
| 2 | 200356 | Saunastunde | 1 | 15,00 | 15,00 |
| Rechnungsbetrag (netto) | | | | | 55,00 |

Hinweis:

Weitere auf Rechnungen übliche Angaben, wie z. B. die Adresse, Mehrwertsteuer, sollen in diesem Entwurf nicht berücksichtigt werden.

Folgende Funktionen sollen verwendet werden:

| | |
|--|--|
| hole_journalsatz() : String | Liest den nächsten Datensatz der Journal-Tabelle in eine Zeichenkette ein. Kann kein Satz mehr gelesen werden, liefert die Funktion den String "". |
| lese_kundenid(satz : String) : Integer | Ermittelt die <i>KundenID</i> innerhalb <i>satz</i> |
| lese_leistungsid(satz : String) : Integer | Ermittelt die <i>LeistungsID</i> innerhalb <i>satz</i> |
| lese_einzelpreis(satz : String) : Double | Liefert den <i>Einzelpreis</i> innerhalb <i>satz</i> |
| lese_anzahl(satz : String) : Integer | Ermittelt die Anzahl (der Leistungen) innerhalb <i>satz</i> |
| hole_bezeichnung(leistungsid : Integer) : String | Liefert die Bezeichnung der Leistung |
| schreibe_kundenid(kundenid : Integer) | Schreibt die <i>KundenID</i> |
| schreibe_kopfzeile() | Schreibt die Kopfzeile der Positionen-Tabelle: Pos LeistungsID Bezeichnung der Leistung Anzahl Einzelpreis Gesamtpreis |
| schreibe_positionenzeile(pos : Integer, leistungsid : Integer, bezeichnung : String, anzahl : Integer, einzelpreis : Double, gesamtpreis : Double) | Schreibt die Zeile für eine Position in der geforderten Darstellung. Der Gesamtpreis einer Position wird von der Prozedur berechnet. |
| schreibe_rechnungssumme(rechnungssumme : Double) | Schreibt die Rechnungssumme |

Entwickeln Sie auf der Folgeseite einen Algorithmus für die Prozedur *erstelleRechnung()*.

Stellen Sie den Algorithmus in Pseudocode, einem Struktogramm oder einem PAP dar.

Codieren Sie das Programm in COBOL, testen und dokumentieren Sie es.