

Übungspaket 5

Abstrakte Programmierung

Übungsziele:

1. Strukturierung einer gegebenen Aufgabe,
2. Bearbeitung von Arbeitsabläufen des alltäglichen Lebens mittels Struktogrammen
3. und der Methode der Schrittweisen Verfeinerung

Skript:

Kapitel: 11 bis 14

Semester:

Wintersemester 2022/23

Betreuer:

Benjamin, Thomas und Ralf

Synopsis:

In diesem Übungspaket betrachten wir Vorgänge aus dem alltäglichen Leben, die letztlich nichts anderes als Algorithmen sind und die wir daher mittels Struktogrammen und der Methode der Schrittweisen Verfeinerung beschreiben wollen. Dabei geht es uns mehr um die Anwendung der erlernten Methoden und *nicht* um eine ultra präzise Beschreibung der alltäglichen Situationen.

Teil I: Stoffwiederholung

Aufgabe 1: Datentypen und Werte

Welche Datentypen werden üblicherweise in der Programmierung verwendet? Gebe zu allen Datentypen ein paar Beispiele für mögliche Werte.

Datentyp	Beispielhafte Werte
Zeichen (<code>char</code>)	'a', 'b', 'X', '8', '.', ' '
Ganzzahlen (<code>integer/int</code>)	-1, 27, 4711, 815, 0
Gleitkommazahlen (<code>double</code>)	123.456, 10e4
Zeichenketten (<code>string</code>)	"Hallo Peter", "Ciao, Inge"

Aufgabe 2: Anweisungen

Welche fünf grundlegenden Anweisungsarten haben wir in Skript und Vorlesung besprochen?

Nr.	Anweisung	Beschreibung
1.	Zuweisung	Variablen einen Wert geben, den man vorher ausrechnet.
2.	Ein- und mehrfache Fallunterscheidung	Um je nach Variablenwert und Ergebnis einer Berechnung unterschiedlich im Programm weiter zu machen.
3.	<code>for</code> -Schleife	Um die selben Anweisungen öfters auszuführen, ohne sie erneut hinschreiben zu müssen. Die Bedingung wird <i>immer vor</i> dem Schleifenanfang überprüft. Im Software Engineering steht die Zahl der Wiederholungen üblicherweise <i>vor</i> dem Eintritt in die Schleife fest.
4.	<code>while</code> -Schleife	Wie oben. Nur geht man im Software Engineering davon aus, dass sich die Zahl der Schleifendurchläufe erst <i>innerhalb</i> der Schleife ergibt.
5.	<code>do-while</code> -Schleife	Wie oben. Aber die Bedingung wird <i>immer</i> am Ende eines Schleifendurchlaufs überprüft. Bei diesem Schleifentyp werden die Anweisungen also mindestens einmal ausgeführt.

Aufgabe 3: Abstrakte Programmierung: warum?

Erkläre mit eigenen Worten, warum wir immer mit der abstrakten Programmierung und nicht gleich mit dem Eintippen eines kleinen C-Programms anfangen.

Die Versuchung ist groß, gleich mit der Kodierung anzufangen. Gerade Anfänger erliegen ihr besonders häufig und scheitern, da sie so gar nicht recht wissen, *was* sie eigentlich programmieren sollen. Daher versuchen wir erst einmal, das *Was* mittels des Konzeptes der abstrakten Programmierung (Schrittweisen Verfeinerung oder Struktogramme) zu klären. Dies geht insbesondere umgangssprachlich sehr gut, da man nicht an eine spezielle Syntax oder Schreibweise gebunden ist. Erst wenn das *Was* geklärt ist, beschäftigen wir uns mit dem *Wie*, also der Umsetzung in eine konkrete Programmiersprache.

Aufgabe 4: Fallunterscheidungen

Beschreibe mit eigenen Worten, warum Fallunterscheidungen notwendig sind und illustriere am Beispiel wie diese in der abstrakten Programmierung aussehen.

Fallunterscheidungen sind immer dann nötig, wenn Ergebnisse oder Zustände geprüft werden müssen, um in Abhängigkeit von diesen weitere Schritte einzuleiten

Raumhelligkeit regeln

```
wenn es ist zu dunkel
dann wenn licht ist aus
    dann Schalte das Licht an
    sonst besorge zusätzliche oder hellere Lichtquellen
sonst Nichts tun, denn es ist hell genug
```

Aufgabe 5: Schleifen

Beschreibe mit eigenen Worten, warum Schleifen notwendig sind und illustriere am Beispiel wie diese in der abstrakten Programmierung aussehen.

Schleifen werden benötigt, um verschiedene Anweisungen und Operation zu wiederholen.

Berechnung von Summe und Mittelwert

```
setze summe = 0
für i = 1 bis Zahl der Messwerte
wiederhole setze summe = summe + Messwert  $x_i$ 
Ausgabe Text "Summe der Messwerte: " Summe
Ausgabe Text "Mittelwert: " Summe / Zahl der Messwerte
```

Teil II: Quiz

Aufgabe 1: Beispiel: Schleife

Was macht nebenstehendes Programm?

Es berechnet die Summe der ganzen Zahlen zwischen -3 und 3 und gibt sie aus.

```
setze summe = 0

für i = -3 bis 3 schrittweite 1
    setze summe = summe + i

Drucke "resultat: " summe
```

Aufgabe 2: Beispiel: Fallunterscheidung

Was macht nebenstehendes Programm?

Wenn i kleiner als j oder j negativ ist, wird i auf den Wert $-j$ gesetzt. In allen anderen Fällen erhält i den Wert $j - i$.

```
Lese Variablen i und j

i > j oder j < 0
wahr           falsch
setze i = -j   setze i = j - i

Drucke "Resultat: " i
```

Teil III: Fehlersuche

Aufgabe 1: Suche von Semantikfehlern

Das folgende Programm soll die Summe aller ungeraden Zahlen von -1 bis 13 (einschließlich) ausrechnen. Doch offensichtlich hat unser Aushilfsprogrammierer DR. NO ein paar Fehler gemacht. Finde die vier Fehler, erkläre und korrigiere sie.

```
setze summe = 0
für i = -3 bis 13 schrittweite 1
    setze summe = summe - i
Drucke "resultat: " summe + 2
```

Fehler	Erläuterung	Korrektur
-3	Die Summenbildung soll bei -1 beginnen	-1
schrittweite 1	Da nur die ungeraden Zahlen genommen werden sollen, muss die Schrittweite 2 genommen werden.	schrittweite 2
-i	Da die <i>Summe</i> berechnet werden soll, muss addiert werden.	+
summe+2	Die Summe wurde bereits in der Schleife berechnet, sodass der Term +2 zu einem Fehler führt.	summe

Eine mögliche Korrektur:

```
setze summe = 0
für i = -1 bis 13 schrittweite 2
    setze summe = summe + i
Drucke "resultat: "summe
```

Teil IV: Anwendungen

Von den folgenden 4 Aufgaben sind mindestens drei zu bearbeiten und jede der beiden Methoden ist mindestens einmal zu verwenden. Bei allen Aufgaben gibt es kein Richtig oder Falsch. Anhand der Aufgaben sollt Ihr lernen, Arbeitsabläufe mittels strukturierter Methoden zu beschreiben und diese in einigen *wenigen* Stufen schrittweise zu verfeinern, bis sie wirklich klar sind. Diesen Prozess könnte man bei jedem Detail nahezu endlos weiterführen; findet also eine sinnvolle Ebene (ein sinnvolles Abstraktionsniveau), bei dem ihr abbrecht, weil die Beschreibung hinlänglich klar ist. Da es uns um die Prinzipien geht, sollte jede Beschreibung aus maximal 15-20 Zeilen bestehen.

Aufgabe 1: Auto betanken

Aufgabenstellung: Ihr fahrt mit dem Auto und der Tank ist fast leer. Beschreibe, wie ihr an die Tankstelle fahrt, tankt und bezahlt.

Benzin tanken

- An die Tankstelle heran fahren
- Zapfsäule aussuchen
- Tankklappe entriegeln
- Benzin einfüllen
- Tankklappe schließen
- Bezahlen
- Wegfahren

An die Tankstelle heran fahren

- Alle Zapfsäulen mit dem richtigen Benzinangebot vergleichen.
- Zapfsäule mit der kürzesten Warteschlange auswählen
- Zur Zapfsäule fahren

Benzin einfüllen

- Zapfpistole aus Halterung nehmen
- Zapfpistole in den Tankeinlass einführen
- Zapfpistole betätigen bis Tank voll ist
- Zapfpistole in die Halterung zurückbringen

Bezahlen

- Je nach Bargeld und Plänen mit Bargeld oder Karte bezahlen
- Quittung einstecken

Aufgabe 2: „Kaffeeklatsch“ in der Kneipe

Aufgabenstellung: Du sitzt mit ein paar Freunden zusammen in der Kneipe, ihr seid am Klönen über dies, das und die Lehrveranstaltung. Je nach Zusammensetzung der Gruppe sollst du verschiedene Getränke bestellen. Auch sollte die Auswahl davon abhängig sein, ob die Schwiegereltern anwesend sind oder nicht. Je nach Stimmungslage sind zwei bis vier Runden zu bestellen.

Kneipenabend

```
setze Rundenzaehler = 0
wiederhole
    setze Rundenzaehler = Rundenzaehler + 1
    wenn Schwiegereltern anwesend
    dann Bestellung: Wein und Bowle
    sonst wenn Frauen anwesend
        dann Bestellung: Sekt
        sonst Bestellung: Bier
bis Rundenzaehler = 4 oder (Stimmung nicht gut und Rundenzaehler >= 2)
```

Stimmung nicht gut

```
Zahl der lachenden Personen < 50%
```

Aufgabe 3: Organisation eines Lernabends

Aufgabenstellung: Für die anstehenden Klausuren trifft ihr euch als Lerngruppe immer reihum bei einem der Mitglieder. Beschreibe, wie du den heutigen Abend organisierst:

Lernabend organisieren

```
Vorbereitungen treffen
wiederhole schaue Sport
solange noch niemand da ist // die anderen schauen auch Sport
wiederhole 1 stunde lernen
    15 minuten pause mit essen und trinken
bis alle muede sind
alle rausschmeissen
```

Vorbereitungen treffen

```
Knabberzeug, Cola und Kaffee einkaufen
Bestell- und Rabattzettel für Pizzadienst organisieren
Fernseh-Fernbedienung verstecken
Aufräumen
```

Aufgabe 4: Reparieren eines Fahrradschlauchs

Aufgabenstellung: Was muss man alles in welcher Reihenfolge machen, wenn man einen kaputten Fahrradschlauch reparieren soll?

Fahrradschlauch reparieren

Rad ausbauen

Mantel mit Mantelhebern abziehen

Schlauch entnehmen

Loch suchen und finden

Loch schließen

Schlauch, Mantel und Rad wieder zusammenbauen

Rad einbauen

Rad aufpumpen

Rad ausbauen

Mit den richtigen Schlüsseln die Muttern lösen

Bremse lösen

wenn Hinterrad

dann Kette abnehmen

Rad aus der Gabel entnehmen

Loch suchen und finden

Schlauch aufpumpen

wenn Loch nicht sofort sichtbar oder hörbar

dann aufgepumpten Schlauch in einen Wassereimer mit Wasser halten

wiederhole Schlauch drehen

bis aufsteigende Blasen sichtbar sind.