



Praktikum zu
**Einführung in die Informatik für
LogWings, WiMas und MedPhys**
Wintersemester 2023/24

Übungsblatt 8

Besprechung:
11.–15.12.2023
(KW 50)

Aufgabe 8.1: Binärsystem

In dieser Aufgabe sollen Sie Zahlen vom Binär- ins Dezimalsystem und umgekehrt umrechnen. In jeder Zeile der unten stehenden Tabelle soll der äquivalente Wert in Binär- und Dezimalschreibweise stehen. Ergänzen Sie die freien Felder der Tabelle entsprechend (es kommen keine negativen Zahlen vor).

Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Dezimalzahl
1	1	0	0	1	1	
1	0	1	0	0	1	
						62
						32

Aufgabe 8.2: Logische Verknüpfungen

- a) Vervollständigen Sie die Wahrheitstabellen für die logischen Verknüpfungen „und“ (\wedge), „oder“ (\vee) und „nicht“ (\neg) mit den Eingängen **a** und **b**.

a	b	$a \wedge b$	$a \vee b$	$\neg a$	$\neg b$
false	false				
false	true				
true	false				
true	true				

- b) Vervollständigen Sie die Wahrheitstabelle für die logische Formel $f = a \wedge c \vee b \wedge \neg b$ mit den Eingängen **a**, **b** und **c**. Versuchen Sie die Formel f vor dem Ausfüllen der Tabelle zu vereinfachen.

a	b	c	$a \wedge c \vee b \wedge \neg b$
false	false	false	
false	false	true	
false	true	false	
false	true	true	
true	false	false	
true	false	true	
true	true	false	
true	true	true	

Aufgabe 8.3: Klammern

Welche Klammer wird wofür verwendet?

i) Kennzeichnung von Blöcken

a) [...]

b) (...)

c) {...}

ii) Parameter-Angabe bei Funktionen

a) [...]

b) (...)

c) {...}

iii) Definition von Arrays und Zugriff auf Arrayelementen

a) [...]

b) (...)

c) {...}

iv) Priorisierung von Berechnungen bzw. Auswertung

a) [...]

b) (...)

c) {...}

Aufgabe 8.4: Funktionsköpfe

Geben Sie einen geeigneten Funktionskopf für die folgenden öffentlichen, statischen Funktionen an.

- a) Eine Funktion **average**, die den Durchschnitt eines **double**-Arrays berechnet und zurückgibt.

- b) Eine Funktion **plus**, die zwei reelle Zahlen miteinander addiert und die Summe zurückgibt.

- c) Eine Funktion **countWords**, die die Wörter in einem **String** zählt und die Anzahl zurückgibt.

- d) Eine Funktion **printMaximum**, die das Maximum eines **int**-Arrays mit `System.out.println` auf dem Bildschirm ausgibt.

- e) Eine Funktion **times**, die einen Integer **n** und einen Integer **x** entgegen nimmt und ein **n** Elemente langes Array, gefüllt mit dem Wert **x** zurückgibt.

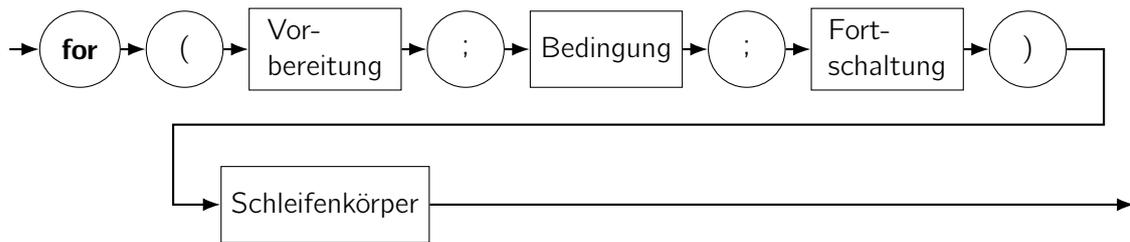
Aufgabe 8.5: Rekursion

Programmieren Sie die Fibonacci-Folge rekursiv. Die Fibonacci-Folge ist wie folgt definiert:

$$\text{fib}(n) = \begin{cases} 0 & \text{wenn } n = 0 \\ 1 & \text{wenn } n = 1 \\ \text{fib}(n-2) + \text{fib}(n-1) & \text{sonst} \end{cases}$$

Aufgabe 8.6: Syntaxdiagramm

Betrachten Sie folgendes Syntaxdiagramm für **for**-Schleifen.



In der linken Spalte (Quelltext) der folgenden Tabelle sehen Sie Programmfragmente. Kennzeichnen Sie durch Ankreuzen, ob diese Fragmente entsprechend des obigen Syntaxdiagrammes gültig oder ungültig sind. Gehen Sie davon aus, dass der Quelltext, der zu den Platzhaltern Vorbereitung, Bedingung, Fortschaltung und Schleifenkörper gehört, fehlerfrei ist.

Quelltext	gültig	ungültig
for (char c='a'; c < 'Z'; c++){}	<input type="radio"/>	<input type="radio"/>
for (int i=0, i-- < 4, i+=2){}	<input type="radio"/>	<input type="radio"/>
for (k = 42; k > 0; k--){}	<input type="radio"/>	<input type="radio"/>
while (i-- < 4) { i+=2; }	<input type="radio"/>	<input type="radio"/>
for { int j=1; j*j < 27513; j*=2}{}	<input type="radio"/>	<input type="radio"/>
for (int i=1; j == 1; i++;) System.out.println();	<input type="radio"/>	<input type="radio"/>
for (int m=0; m < 9; m++) while (m != 10) { m++; }	<input type="radio"/>	<input type="radio"/>
do { m++; } while (m != 10);	<input type="radio"/>	<input type="radio"/>

Aufgabe 8.7: Schleifen

- a) Ersetzen Sie das folgende Programmfragment gleichwertig, sodass anstelle einer do-while-Schleife eine while-Schleife verwendet wird.

```
do{
    a = b - i;
    System.out.println(a);
} while (a > 0);
```

- b) Ersetzen Sie das folgende Programmfragment gleichwertig, sodass anstelle der while-Schleifen zwei for-Schleifen verwendet werden.

```
int i = 10;
while (i > 5) {
    int x = 0 ;
    while (x < 5) {
        System.out.println(i + ":" + x);
        x = x + 2;
    }
    i = i - 2;
}
```

Aufgabe 8.8: Muster

Schreiben Sie öffentliche, statische Funktionen, die die folgenden Muster erzeugen. Für jedes Muster müssen zwei ineinander verschachtelte for-Schleifen verwendet werden. Die Schleifen müssen maßgeblich an der Mustererzeugung beteiligt sein. Ausgaben außerhalb der Schleifen, wie auch die Verwendung von Strings sind **nicht** zulässig.

Tipp: Sie können die Funktion `Math.pow(b, e)` zur Berechnung von b^e nutzen. Denken Sie an die explizite Typkonvertierung von `double` nach `int`.

a)

```
*_*_*_*_*_*
*_*_*_*_*_*
*_*_*_*_*_*
*_*_*_*_*_*
*_*_*_*_*_*
```

b)

```
1
12
124
1248
```

Aufgabe 8.9: Arrays

Schreiben Sie eine Funktion, die jedes Element eines übergebenen int-Arrays mit einer ebenfalls übergebenen ganzen Zahl multipliziert und das Ergebnis zurückgibt.