

Praktikum zu
**Einführung in die Informatik für
LogWilngs und WiMas**
Wintersemester 2015/16

Übungsblatt 4

Besprechung:
23–27.11.15
(KW 48)

Vorbereitende Aufgaben

Aufgabe 4.1: Zeitumrechnung Vorbereitung

Ziel dieses Übungszettels ist es, ein Verständnis für die Ganzzahldivision und große Zahlen zu entwickeln. Sie werden dazu schrittweise angeleitet, ein Programm zur Ausgabe einer Uhrzeit zu schreiben.

In vielen Rechnersystemen wird Zeit in vergangenen **Sekunden** seit dem 1.1.1970 angegeben. Der Zeitpunkt 42 z. B. repräsentiert den 1.1.1970 00:00:42, der Zeitpunkt $86.400 = 24 \cdot 60 \cdot 60$ den 2.1.1970 00:00:00 und der Zeitpunkt 554.992.367 den 3.8.1987 14:32:47.

Lösen Sie nun die folgenden Aufgaben!

- Wie viele **Tage** sind zum Zeitpunkt 3.888.000 vergangen?

- Wie viele **Tage** sind zum Zeitpunkt 630.720.000 vergangen?

- Repräsentiert dieser Zeitpunkt einen 1. Januar? Wenn nicht: Warum?

- Wenn der Zeitpunkt 1.446.708.600 der 05.11.2015 07:30:00 ist, welches Datum hat dann der Zeitpunkt 1.448.265.600?

Präsenzaufgaben

Aufgabe 4.2: Wiederholung

In dieser Aufgabe sollen Sie die bisher gelernten Operationen (Kapitel 3.1, Folie 5–18) wiederholen.

- Wie deklarieren Sie eine Variable vom Typ **float** mit dem Namen **velocity**?

- Wie weisen Sie einer bereits deklarierten Variablen mit dem Namen **value** den Wert 25 zu?

- Wie deklarieren Sie eine Variable vom Typ **int** mit dem Namen **sum** und initialisieren diese mit der Summe zweier bereits deklarierten und initialisierten Variablen **x** und **y**?

Aufgabe 4.3: Zeitumrechnung Implementierung

Wir werden nun die in Aufgabe 1 berechneten Ergebnisse nutzen, um ein Programm zu formulieren.

- (a) – Legen Sie eine neue Klassendatei mit dem Namen **Clock** an.
– Passen Sie die Datei an, sodass sie diesem Grundgerüst entspricht:

```
1 package blatt04;
2
3 public class Clock {
4
5     public static void main(String[] args) {
6         /* Anfang des Programmcodes */
7
8         /* Ende des Programmcodes */
9     }
10 }
```

- Das Programm soll nun innerhalb des durch Kommentare markierten Bereiches implementiert werden.

- (b) – Um eine Uhrzeit zu repräsentieren, benötigen Sie 3 Werte: Stunden, Minuten und Sekunden. Verwenden Sie dafür 3 Variablen mit den Namen **seconds**, **minutes** und **hours**.
- Zusätzlich möchten wir den Zeitpunkt (Anzahl der verstrichenen Sekunden) in einer Variable mit dem Namen **time** vermerken.
- Deklarieren Sie deshalb insgesamt 4 **long** Variablen, in denen diese Werte gespeichert werden sollen.
- Wenn Sie diese Werte als **int** deklarieren würden, welches Problem kann in der Zukunft auftreten?
-
-

- (c) – Initialisieren Sie hinter den Deklarationen den Wert des Zeitpunkts mit einem beliebigen Wert.
- Weisen Sie anschließend den Variablen für Stunde, Minute und Sekunde **in Abhängigkeit zum Zeitpunkt** den passenden Wert durch eine Berechnung zu.
- (d) – Zu guter Letzt wollen wir die berechneten Werte nutzen, um die Zeit auszugeben.
- Rufen Sie die bekannte Systemfunktion **System.out.println()** auf und geben Sie einen informativen Text aus, der sowohl den gewählten Zeitpunkt als auch die Uhrzeit ausgibt.
- Testen Sie das Programm nun mit einem Klick auf den **Run**-Button, wie es auf dem ersten Übungsblatt beschrieben wurde.

Aufgabe 4.4: Bedingte Anweisungen – Bedingungen

In dieser Aufgabe sollen Sie sich mit der Auswertung boolescher Ausdrücke beschäftigen.

Vervollständigen Sie folgende Tabelle:

Ausdruck	boolescher Wert
$5 > 6$	false
<code>true && false</code>	
<code>true false</code>	
$(5 < 8) \ \&\& \ (3 > 1)$	
$17 < 11 \ \ (1 > 2 \ \&\& \ \text{true})$	
<code>true false && false</code>	
<code>true && false false</code>	
$5 * 8 > 22 \ \&\& \ 8 < 3 \ \ 3 < 2 \ \&\& \ 36 == 3 * 12$	

Aufgabe 4.5: Schaltjahre

In dieser Aufgabe sollen Sie ein kurzes Programm schreiben, das entscheidet, ob ein gegebenes Jahr ein Schaltjahr ist oder nicht.

- Erstellen Sie eine neue Klasse namens **LeapYear** im **blatt04** Paket.
- Ergänzen Sie das Grundgerüst:

```
1 package blatt04;
2
3 public class LeapYear {
4     public static void main(String[] args) {
5         /* ... */
6     }
7 }
```

- Deklarieren und initialisieren Sie eine Variable vom Typ **int** mit dem Namen **year** und einem beliebigen Wert, der eine Jahreszahl repräsentieren soll.
- Ein Jahr ist ein Schaltjahr, wenn es durch 4 Teilbar ist, außer es ist durch 100 und nicht durch 400 Teilbar.
- Welcher boolesche Ausdruck repräsentiert diese Eigenschaft in Abhängigkeit zur Variablen **year** ?

-
- Implementieren Sie eine bedingte Anweisung, die den Text „Leap Year“ ausgibt, wenn die gegebene Jahreszahl ein Schaltjahr repräsentiert und „No Leap Year“ ausgibt, wenn die gegebene Jahreszahl kein Schaltjahr repräsentiert.

Aufgabe 4.6: Bedingte Anweisungen – Fallunterscheidungen

In dieser Aufgabe sollen Sie sich mit dem Zusammenhang zwischen bedingten Anweisungen und Fallunterscheidungen beschäftigen. (Kapitel 3.2 Folie 16–22)

Gegeben ist folgendes Programmfragment einer bedingten Anweisung. Gehen Sie davon aus, dass die **int** Variable **i** bereits initialisiert ist:

```
if(i == 0) {
    System.out.println("zero");
} else if(i == 1) {
    System.out.println("one");
} else if(i == 2) {
    System.out.println("two");
} else if(i == 3) {
    System.out.println("three");
} else if(i == 4) {
    System.out.println("four");
} else if(i == 5) {
    System.out.println("five");
} else if(i == 6) {
    System.out.println("six");
} else if(i == 7) {
    System.out.println("Between seven and nine");
} else if(i == 8) {
    System.out.println("Between seven and nine");
} else if(i == 9) {
    System.out.println("Between seven and nine");
} else {
    System.out.println("Not between one and nine");
}
```

Geben Sie eine Fallunterscheidung mit den Schlüsselwörtern **switch**, **case**, **default** und **break** an, die die gleiche Funktionalität umsetzt.

Ergänzende Aufgaben

Aufgabe 4.7: Monatsnamen und Stringvariablen

In dieser Aufgabe sollen Sie Fallunterscheidungen näher betrachten und eine Variable vom Typ **String** verwenden.

Implementieren Sie ein Programm namens **Monthname**, das zu einer gegebenen Zahl den entsprechenden Monatsnamen des Gregorianischen Kalenders ausgibt:

- Erstellen Sie im Paket **blatt04** eine neue Klasse namens **Monthname**.
- Ergänzen Sie den Rumpf der Klasse wie gehabt um die **main**-Methode.
- Deklarieren Sie eine Variable mit dem Namen **month** und initialisieren Sie diese mit einem beliebigen Wert, möglichst zwischen 1 und 12.
- Deklarieren Sie eine weitere Variable mit dem Namen **message** und dem Typ **String**.
- Implementieren Sie eine Fallunterscheidung mit den Schlüsselwörtern **switch**, **case**, **break** und **default**, um der Variablen **message** den entsprechenden Monatsnamen zuzuweisen.
- Im **default**-Fall sollen Sie der Variablen **message** eine Fehlermeldung zuweisen, die den Benutzer darüber informiert, dass zur gegebenen Zahl kein Monatsname existiert.
- Geben Sie hinter der Fallunterscheidung den Inhalt der Variablen **message** mit einer `System.out.println()`-Anweisung aus.