

Praktikum zu
**Einführung in die Informatik für
LogWilngs und WiMas**
Wintersemester 2017/18

Übungsblatt 4
Besprechung:
13.–17.11.2017
(KW 46)

Vorbereitende Aufgaben

Aufgabe 4.1: Zeitumrechnung – Vorbereitung

Ziel dieser Aufgabe und Aufgabe 3 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 werden Zeitpunkte in vergangenen **Sekunden** seit dem 1.1.1970 angegeben, diese nennt man **Zeitstempel** oder auch UNIX-Zeitstempel¹.

Der Zeitstempel 42 z. B. repräsentiert den 1.1.1970 00:00:42, der Zeitstempel $86.400 = 24 \cdot 60 \cdot 60$ den 2.1.1970 00:00:00 und der Zeitstempel 554.992.367 den 3.8.1987 12:32:47.

Lösen Sie nun die folgenden Aufgaben:

- Wie viele ganze Tage sind zum Zeitstempel 3.888.000 vergangen?

- Wie viele ganze Tage sind zum Zeitstempel 630.720.000 vergangen?

- Wo steht der Sekundenzeiger einer Uhr zum Zeitstempel 1428866372?
Wo der Minutenzeiger?

- Wie viele Sekunden seit Tagesbeginn sind zum Zeitstempel 1030637317 vergangen?

¹Auf diesem Übungszettel betrachten wir die Zeitzone nicht! Beachten Sie, wenn Sie Ihre Berechnungen maschinell überprüfen, dass UNIX-Zeitstempel in UTC angegeben sind.

Präsenzaufgaben

Aufgabe 4.2: Variablen – Wiederholung

In dieser Aufgabe sollen Sie den Umgang mit Variablen (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 Variable 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 Zeitstempel (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 die Variable **time** als **int** deklarieren würden, welches Problem kann in der Zukunft auftreten?
-
-

- c)
- Initialisieren Sie hinter den Deklarationen den Wert des Zeitstempels mit einem beliebigen Wert.
 - Weisen Sie anschließend den Variablen für Stunde, Minute und Sekunde **in Abhängigkeit zum Zeitstempel** 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 Zeitstempel 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.
- e)
- Was für Schwierigkeiten gäbe es, würden wir **seconds**, **minutes** und **hours** als **int** deklarieren?
-
-

Aufgabe 4.4: if-Anweisungen – Bedingungen

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

Vervollständigen Sie die folgende Tabelle, indem Sie die folgenden Ausdrücke auswerten:

Ausdruck	boolescher Wert
5 > 6	false
true && false	
true false	
(5 < 8) && (3 > 1)	
17 < 11 (1 > 2 && true)	
true false && false	
true && false false	
5 * 8 > 22 && 8 < 3 3 < 2	

Aufgabe 4.5: if-Anweisungen und switch-Anweisungen

In dieser Aufgabe schreiben Sie zum ersten Mal eine **if**-Anweisung, die Sie später in eine **switch**-Anweisung umwandeln werden.

Wir wollen ein Programm schreiben, das Zahlen in Worten ausschreibt. Dabei betrachten wir nur natürliche Zahlen (inklusive 0). Das wollen wir schrittweise entwickeln.

- a)
 - Erstellen Sie eine neue Klasse **IntToText** und legen Sie in der main-Methode eine **int**-Variable mit dem Namen **number** an.
 - Fragen Sie in einer **if**-Anweisung ab, ob **number** den Wert 0 hat und geben Sie in diesem Fall „zero“ aus.
 - Initialisieren Sie **number** mit verschiedenen Werten und testen Sie, ob genau dann, wenn **number** den Wert 0 hat, der Text ausgegeben wird.
 - b)
 - Erweitern Sie Ihr Programm um eine if-Anweisung, die „one“ ausgibt, wenn **number** den Wert 1 hat.
 - Muss diese if-Anweisung vor oder hinter der if-Anweisung aus Aufgabenteil a) stehen?
-
- c)
 - Erweitern Sie ihr Programm um die passenden Ausgaben bei den Werten 2–4.
 - d)
 - Des Weiteren interessiert uns, ob der Wert zwischen 5 und 7 oder größer als 7 ist.
 - Geben Sie im Falle, dass der Wert **number** zwischen 5 und 7 ist, den Text „between five and seven“ aus.
 - Ist der Wert größer als 7, soll „greater than seven“ ausgegeben werden.
 - e)
 - Wandeln Sie diese Ansammlung von **if**-Anweisungen in eine **switch**-Anweisung um.

Ergänzende Aufgaben

Aufgabe 4.6: Zahlen-Ausschreibung

Betrachten Sie Ihren Code aus Aufgabe 5. Welche Änderungen sind nötig, um die Zahlen bis 99 richtig auszuschreiben?
