

Praktikum zu
**Einführung in die Informatik für
LogWings und WiMas**
Wintersemester 2011/12

Übungsblatt 3

Bearbeitungszeit:
07.–11.11.2011

Hinweis: Auf diesem Übungsblatt müssen Sie in einigen Aufgaben bestimmte Webseiten öffnen. Wenn Sie die entsprechenden URLs nicht abschreiben wollen, können Sie die jeweilige Webseite öffnen, indem Sie in dem PDF dieses Übungsblatts ihre URL anklicken.

Aufgabe 3.1 – Fehlersuche

In dieser Aufgabe sollen Sie nicht programmieren. Bearbeiten Sie stattdessen die Aufgabe nur auf dem Papier – **so, wie Sie es auch in der Klausur tun müssen.**

- Betrachten Sie das untenstehende Java-Programm. Was soll es Ihrer Meinung nach ausgeben?

- Das Programm enthält **Syntaxfehler**, die dazu führen, dass das Programm nicht kompiliert werden kann. Unterstreichen Sie die Syntaxfehler im Programmtext und korrigieren Sie sie auf den entsprechenden Linien daneben.
- Werden **Laufzeitfehler** auftreten, wenn das Programm nach der Korrektur der Syntaxfehler ausgeführt wird? Wenn ja, welche Laufzeitfehler erwarten Sie?

- Enthält das Programm außerdem **Semantikfehler**, die dazu führen, dass das Programm nicht das berechnet, was von ihm erwartet wird? Wenn ja: Wie müssen die Fehler korrigiert werden?

Der Programmtext:

```
1 Class WochenImSchaltjahr {  
2     public static void main (string[] args) {  
3         integer tage;  
4         int wochen;  
5         int resttage;  
6  
7         tage = 366  
8         wochen = tage * 7;  
9         resttage = tage % 7;  
10  
11     System.out.pint("Ein Schaltjahr besteht");  
12     //System.out.print(" manchmal");  
13     Systemout.println(" aus " + wochen +  
14         " Wochen und " resttage + " Tagen.);  
15     {  
16     }
```

Aufgabe 3.2 – Die Java-VM

In Kapitel 1 (Folien 6, 7) der Vorlesung wurde das allgemeine Konzept einer „virtuellen Maschine“ (VM) vorgestellt. In dieser Aufgabe sollen Sie nun Informationen speziell über die „Java-VM“ und die „Java-API“ sammeln. Lesen Sie dazu folgenden Text auf den Webseiten der Firma Oracle:

<http://download.oracle.com/javase/tutorial/getStarted/intro/definition.html>

Bearbeiten Sie dann die folgenden Fragen:

- Beschreiben Sie noch einmal in eigenen Worten, was beim Kompilieren eines Java-Programms geschieht.

- Wie lautet der Name des Java-Kompilierprogramms?

- Woraus besteht eine `.class`-Datei?

- Was ist der Vorteil, den die Java-VM im Hinblick auf verschiedene Betriebssystem-Plattformen bietet?

- Können Sie demnach eine `.class`-Datei, die Sie an der Uni unter dem Betriebssystem Linux kompiliert haben auch zu Hause in Ihrer Java-VM unter einem anderen Betriebssystem ausführen?

- Was ist der Nachteil der in der VM ausgeführten Java-Programme im Vergleich zu betriebssystemabhängigen (sogenannten „nativen“) Programmen, wie sie zum Beispiel mit der Programmiersprache C erstellt werden?

- Wofür steht die Abkürzung API?

- Woraus besteht die Java-API?

Falls Sie für zukünftigen Aufgaben etwas nachschlagen müssen, finden Sie die vollständige Java API unter folgender Adresse: <http://download.oracle.com/javase/6/docs/api/>

Aufgabe 3.3 – Einlesen von der Tastatur

In dieser Aufgabe werden Sie ein Programm schreiben, das zwei Zahlen vom Benutzer einlesen kann.

Schreiben Sie zunächst folgenden Quellcode ab und führen Sie das Programm aus.

```
1 import java.util.Scanner;
2
3 class Einlesen {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Bitte geben Sie eine Zahl ein");
7         int x = scanner.nextInt();
8         System.out.println("Ihre Eingabe war: " + x);
9     }
10 }
```

Bis auf die Zeilen 1, 5 und 7 sollte Ihnen alles bekannt sein. In Zeile 1 teilen Sie dem Compiler mit, dass eine *Klasse*, die den Namen `Scanner` hat, aus der Java-API eingebunden werden soll. Was genau eine Klasse ist, erfahren Sie in den nächsten Wochen. Im Moment ist wichtig, dass der `Scanner` verschiedene Datentypen von der Tastatur einlesen kann, insbesondere auch Zahlen vom Typ `int`. In Zeile 5 wird der `Scanner` „gestartet“, oder auch *initialisiert*. Auch dies wird erst später genauer erklärt.

In Zeile 7 wird der `Scanner` benutzt: Sobald das Programm an die Stelle kommt, an der `scanner.nextInt()` steht, wird eine Zahl vom Benutzer abgefragt und kann vom Programm weiterverwendet werden.

Erweitern Sie das Programm so, dass zwei Zahlen abgefragt werden und anschließend ihre Summe und ihre Differenz ausgegeben werden.

Aufgabe 3.4 – Deklaration, Initialisierung und Zuweisung

Schauen Sie sich folgendes Programm an und überlegen Sie, was die Ausgabe sein könnte.

```
1 class DeklarationBeispiel {
2     public static void main(String[] args) {
3         int a;
4         a = 7;
5         int b = 5;
6         System.out.println("a: " + a + ", b: " + b);
7         int c = a + 3;
8         System.out.println("c: " + c);
9         a = b;
10        System.out.println("a: " + a + ", c: " + c);
11    }
12 }
```

Worin unterscheiden sich die Zeilen 3-4 von der Zeile 5?

Notieren Sie die von Ihnen vermutete Ausgabe:

Führen das Programm nun in Java aus (nachdem Sie es abgeschrieben haben) und überprüfen Sie Ihre Vermutung. Beschreiben Sie kurz, was das Programm tut:

Aufgabe 3.5 – Typ Boolean

Schauen Sie sich folgendes Programm an und überlegen Sie, was die Ausgabe sein könnte.

```
1 class BooleanBeispiel {
2     public static void main(String[] args) {
3         boolean a = 7 > 6;
4         boolean b = -5 < -10;
5         System.out.println("a: " + a + ", b: " + b);
6         boolean c = 11 == 11;
7         System.out.println("c: " + c);
8         System.out.println("a==b = " + (a==b));
9     }
10 }
```

Notieren Sie die von Ihnen vermutete Ausgabe:

Führen das Programm nun in Java aus (nachdem Sie es abgeschrieben haben) und überprüfen Sie Ihre Vermutung. Beschreiben Sie kurz, was das Programm tut:

Aufgabe 3.6 – Teilbarkeit

Verwenden Sie das in den Aufgaben 3.3 und 3.5 Gelernte und schreiben Sie ein Programm, das folgendes leistet:

- Den Benutzer auffordern eine ganze Zahl einzugeben.
- Die Zahl einlesen.
- Feststellen, ob die eingegebene Zahl durch 3 teilbar ist und das Ergebnis in einer Variable vom Typ `boolean` speichern.
- Das Ergebnis ausgeben.

Ergänzende Aufgabe

Aufgabe 3.7 – Zahlensysteme

In dieser Aufgabe solle Sie ein Programm zur Umrechnung von Binärzahlen in Dezimalzahlen schreiben. Konkret sollen Sie ein Programm schreiben, das vom Benutzer vier mal jeweils ein Bit (0 oder 1) einliest und die resultierende Dezimalzahl ausgibt.

Betrachten Sie folgende Tabelle und testen Sie zunächst, ob Ihr Programm die in den ersten beiden Zeilen angegebenen Zahlen korrekt ausrechnet. Fügen Sie dann die fehlenden Werte ein.

Bit 3	Bit 2	Bit 1	Bit 0	Dezimalzahl
0	1	0	1	5
1	1	0	0	12
0	0	1	0	
1	1	1	1	
1	0	0	1	