

Aufgaben Programmierkurs

Übungsblatt 1



TECHNISCHE
UNIVERSITÄT
DARMSTADT

von Uli Fahrer & Dennis Albrecht

Wintersemester 2014/2015

Aufgabe 1 Multiple Choice

Kreuze zu jeder Antwort an, ob sie zutrifft (w) oder nicht (f).

Aufgabe 1.1 Theoriefragen (Stufe 1)

w f

- ☐ ☐ Ein Compiler führt ein Programm aus.
- ☐ ☐ Ein Compiler wandelt ein Programm in eine andere Darstellung um.
- ☐ ☐ Java ist eine interpretierte Sprache.
- ☐ ☐ Java ist eine compilierte Sprache.
- ☐ ☐ Java wird sowohl compiliert als auch interpretiert.
- ☐ ☐ Compilierte Java-Programme können immer nur auf dem eigenen Rechner ausgeführt werden.

Aufgabe 1.2 Fehlerarten (Stufe 2)

w f

- ☐ ☐ Programme mit lexikalischen Fehlern können compiliert werden.
- ☐ ☐ Programme mit syntaktischen Fehlern können compiliert werden.
- ☐ ☐ Programme mit semantischen Fehlern können compiliert werden.

Überlege auch, ob Fehler, welche sich beim compilieren nicht bemerkbar machen, bei Programmausführung **zwingend** zu Fehlern führen.

Aufgabe 2 Zum warm werden

Aufgabe 2.1 Hello World Reloaded (Stufe 1)

Erweitere das in der Vorlesung kennen gelernte Hello-World-Programm um beliebige Ausgaben deiner Wahl. Mache dabei auch Gebrauch von Kommentaren und beantworte die folgenden Fragen:

- a) Was ist das besondere an der Mainfunktion?
- b) Ändere den Dateinamen deines Quellprogramms und versuche das Programm erneut zu compilieren. Was passiert und warum?

Aufgabe 2.2 Finde den Fehler (Stufe 1)

Gegeben ist folgendes Listing. Finde alle Fehler und überlege dir, zu welchem Fehlertypen die Fehler gehören!

```
1 public class Error_Listing
2     public static void main(String[] args){
3         system.ou.print(Hello World)
4     }
5 }
```

Aufgabe 2.3 Textausgaben (Stufe 1)

Es gibt in Java zwei Möglichkeiten (Befehle) um Texte einfach und schnell auszugeben.

a) Was ist der Unterschied/sind die Unterschiede zwischen den beiden Befehlen `System.out.print` und `System.out.println`?

b) Was bedeutet das für die folgenden Befehlsfolgen?

1. `System.out.print("To_be_")`
`System.out.print("or_not_")`
`System.out.print("to_be?")`

2. `System.out.println("To_be_")`
`System.out.println("or_not_")`
`System.out.println("to_be?")`

3. `System.out.print("To_be_")`
`System.out.print("or_not_")`
`System.out.println("to_be?")`

4. `System.out.println("To_be_")`
`System.out.print("or_not_")`
`System.out.println("to_be?")`

Aufgabe 3 Variablen

Aufgabe 3.1 Namenskonvention (Stufe 1)

Welches sind **gültige** Variablennamen?

a) Matrikelnummer

c) `_meinAlter`

e) `$Dollar`

b) `ÄhmKeineAhnung`

d) `&Richtig`

f) `5teVariable`

Aufgabe 3.2 Variablenwerte (Stufe 2)

Welche Zuweisungen compilieren und werden ausgeführt und welche Werte wurden (wenn möglich) in den Variablen gespeichert? Die Anweisungen werden nacheinander ausgeführt. So bezeichnet *c* in der siebten Zuweisung das Ergebnis der dritten Zuweisung.

- | | | | |
|-------------------------------|---|---------------------------------|---|
| • <code>int a;</code> | : | • <code>int g = c;</code> | : |
| • <code>float b = 42;</code> | : | • <code>double h = b;</code> | : |
| • <code>int c = 23;</code> | : | • <code>int i = b;</code> | : |
| • <code>char d = 73;</code> | : | • <code>int j = (int) b;</code> | : |
| • <code>float e = 5.7;</code> | : | • <code>double k = j/g;</code> | : |
| • <code>int f = 9.81f;</code> | : | • <code>int l = d;</code> | : |

Aufgabe 3.3 Case-sensitive (Stufe 1)

Gegeben ist folgendes Listing:

```
1 int meinAlter = 21;
2 MeinAlter = MeinAlter + 1;
3 System.out.println(meinAlter);
```

Beim Compilieren der zweiten Zeile wirft der Compiler einen Fehler. Warum? Verbessern Sie das Listing.

Aufgabe 3.4 the awesome 42 (Stufe 2)

Deklariere eine Integer Variable und initialisiere sie mit dem Wert 42.5. Warum gibt es Compilerfehler? Erkläre dieses Verhalten.

Aufgabe 3.5 Zahlentypen (Stufe 3)

- Deklariere und initialisiere eine double Konstante für PI (Wert 3,14159...).
- Deklariere eine float Variable und initialisiere sie mit dem Wert der Pi-Konstanten. Denke daran, dass hier eine Typverengung stattfindet, die der Compiler nicht ohne weiteres akzeptiert.
- Deklariere eine Variable vom Typ Byte und initialisiere diese mit dem Dezimalwert 127. Inkrementiere anschließend die Variable. Gib nun den Wert der Variable auf der Konsole aus. Was stellst du fest? Erkläre was passiert ist. Halte das Ergebnisse in einem Kommentar fest.

Aufgabe 3.6 Ein- und Ausgabe (Stufe 2)

Im Rahmen der Vorlesung hast du die Klasse `Input` kennengelernt, welche es dir erlaubt, auf einfache Art und Weise Benutzereingaben von der Konsole einzulesen. Schreibe jetzt ein Programm, welches mit Hilfe der `Input`-Klasse nacheinander folgende Eingaben einliest und in Variablen mit jeweils passendem Datentyp speichert.

- deinen Namen als Zeichenkette
- dein Geburtsdatum als drei ganze Zahlen: Tag, Monat, Jahr.

Insgesamt sollen also vier Eingaben verarbeitet werden. Achte dabei darauf dass dein Programm die erwarteten Eingaben textuell klar strukturiert (Stichwort Bedienerfreundlichkeit).

Nachdem die letzte Eingabe verarbeitet ist, soll folgende Zeichenkette ausgegeben werden (wobei natürlich die Platzhalter ersetzt werden sollen):

Ausgabe: "<dein Name> hat am <tt.mm.jjjj> Geburtstag."

Hinweis: Die Verarbeitung von Eingaben wird im Foliensatz 'Variablen.pdf' erläutert.

Aufgabe 4 Arithmetische Grundlagen

Aufgabe 4.1 Quadratische Funktionen (Stufe 3)

Gegeben sind die Funktionen $f(x) = x^2 - 6x + 5$ und $g(x) = -x^2 + 2x + 3$.

Berechne die Nullstellen der polynomiellen Funktionen $f(x)$ und $g(x)$ und speichere die Ergebnisse in einer geeigneten Variable.

Hinweise:

- Sei $f(x) = x^2 + px + q$. PQ-Formel $x_{1,2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$
- Wurzelfunktion: `Math.sqrt(operand)`
- Potenzieren: `Math.pow(base, exponent)`

Aufgabe 4.2 Gaußsche Summenformel (Stufe 3)

Die **gaußsche Summenformel** ist eine Formel für die Summe der ersten n aufeinander folgenden natürlichen Zahlen \mathbb{N} . Sie ist gegeben durch:

$$1 + 2 + 3 + \dots + n = \sum_{k=1}^n k = \frac{n \cdot (n + 1)}{2}$$

- Berechne die Summe der ersten **100** natürlichen Zahlen und speichere das Ergebnis in einer Variable **sum**. Wähle dafür einen geeigneten Datentyp!
- Gib dieses Ergebnis anschließend auf der Konsole aus.

Hinweis: Verwende die geschlossene Form der Summe (den Bruch rechts des Gleichheitszeichens).