

## Aufgabenblatt 2

Arbeiten Sie im Lehrbuch<sup>1</sup> Abschnitt 1.6, Kapitel 2 und Anhang C bis inklusive C.3 durch, bis Sie den Inhalt vollständig verstanden haben. Lösen Sie anschließend die unten angegebenen Aufgaben. Für die Programmieraufgabe benötigen Sie die Dateien `In.class` und `Out.class`. Sollten Sie die Entwicklungsumgebung „Eclipse“ einsetzen, sollten Sie dagegen die Dateien `In.java` und `Out.java` nutzen. Sie finden die Dateien unter:

<http://www.ssw.jku.at/JavaBuch/>

Speichern Sie die Dateien im gleichen Verzeichnis wie Ihre eigenen Programme, wobei es je nach Programmierungsumgebung Alternativen geben kann. Bei Problemen mit der Verwendung der Klassen können Sie Teilpunkte erreichen, wenn Sie bei der Ausgabe statt `Out.println` zunächst `System.out.println` schreiben und statt der Eingabe die Werte im Programm an Variablen zuweisen.

a) Geben Sie alle Buchstabenfolgen an, die von der folgenden Grammatik erzeugt werden:

Buchstabenfolge = [ "a" | {"b"} "c" "d" ] "e" .

(5 Punkte)

b) Betrachten Sie die Grammatik zur Definition von Gleitkommazahlen auf Seite 18 des Lehrbuchs. Diese ist gegenüber Java vereinfacht. Sind

- $.73E5$
- $13.8E + 08$
- $17E - 3$
- $0.0E0$

gültige Gleitkommazahlen entsprechend *dieser* Grammatik? Begründen Sie ausführlich. Für die Gleitkommazahlen muss dazu eine Herleitung angegeben werden. Schreiben Sie diese in ähnlicher Form wie auf Folie 23 oder 24 auf.

(8 Punkte)

c) Handelt es sich bei

```
int myChar = 'x';
```

um eine Deklaration, Initialisierung und/oder Zuweisung (ohne Begründung) ?

(1 Punkt)

---

<sup>1</sup>H. Mössenböck: Sprechen Sie Java? dpunkt-Verlag, 2014

d) Handelt es sich bei

`x++`

um einen Ausdruck, eine Anweisung und/oder eine Ausdrucksanweisung (ohne Begründung)?

(1 Punkt)

e) Das folgende Java-Programm enthält mehrere Fehler. Finden Sie diese und geben Sie jeweils an, ob es sich um einen Syntaxfehler, Typfehler, Fehler im Programmierstil oder sonstigen Fehler handelt.

```
class wrong {  
  
    static final long MUCH = 10000;  
  
    public static void main(String[] args) {  
  
        int dangerous = 13;  
        int Zero = '0';  
        int quiteMuch = MUCH / 2; /* rundet ab  
        dangerous = dangerous--;  
        int i = 1;  
        int j = 5 / (i-1);  
        int i++; // i erhoeihen  
    }  
}
```

(6 Punkte)

f) Korrigieren Sie das Programm aus Aufgabe e), so dass alle Fehler beseitigt werden. Nehmen Sie dabei nicht mehr Änderungen als notwendig vor. Im Anschluss muss sich das Programm korrekt übersetzen und ausführen lassen. Hinweis: Es gibt mehrere richtige Lösungen, die sich in ihrer Funktionalität unterscheiden.

(4 Punkte)

g) Im folgenden Programmabschnitt sind a und b vom Typ int. Vereinfachen Sie den Programmabschnitt so weit wie möglich:

```
a = b;  
a -= (b--) + 1;
```

(2 Punkte)

h) Es sei

```
long g = 2L;  
int i = 3;  
short s = 4;
```

Geben Sie *alle* impliziten Typkonvertierungen an, die bei der Berechnung des folgenden Ausdrucks ausgeführt werden. Beschreiben Sie jeweils genau, welcher Wert bzw. welches Zwischenergebnis von welchem in welchen Typ konvertiert wird.

```
g = (s / 4 + g) - i;
```

(3 Punkte)

i) Fritzchen.java: Fritzchen hat in seiner Sparbüchse eine beträchtliche Anzahl von Münzen zu 5, 2 und 1 Cent gespart. Diese hat er getrennt gezählt und möchte nun wissen, wieviele Euro er besitzt. Schreiben Sie ein Java-Programm, das ihm dabei hilft. Das Programm soll sich folgendermaßen verhalten:

```
Anzahl 5-Cent Muenzen: 31  
Anzahl 2-Cent Muenzen: 22  
Anzahl 1-Cent Muenzen: 5  
Du besitzt 2 Euro und 4 Cent.
```

Achten Sie auf den Programmierstil (Namenswahl, Groß-/Kleinschreibung, Einrückung etc.) sowie auf eine übersichtliche Ausgabe (z.B. Verwendung von Leerzeichen, Zeilenwechsel am Programmende). In Ihrer Lösung dürfen Sie nur die bereits behandelten Java-Konstrukte verwenden, also keine Verzweigungen, Gleitkommazahlen, etc. Benennen Sie Ihre Java-Klasse wie in der Aufgabenstellung angegeben (also Fritzchen) und geben Sie den Quelltext (also die .java-Datei), *nicht* die .class Datei, ab.

(10 Punkte)

**Abgabetermin:** Die Lösungen sind bis spätestens Donnerstag, den 2.11.2017 um 8:00 Uhr (strikt!) über das elektronische Abgabesystem einzureichen. Nachträglich eingereichte Lösungen zählen als nicht abgegeben.