

Lösungsvorschlag zur Probeklausur Programmierung für Alle (Java), Dezember 2006

Dieser Lösungsvorschlag erhebt keinen Anspruch auf Korrektheit und Vollständigkeit. Bei Fehlern oder sonstigen Anmerkungen bitte eine Email an stefan.kleeschulte@rwth-aachen.de.

Aufgabe 1

Verständnisaufgaben

- (a) Ein Algorithmus ist eine Berechnungsvorschrift zur Lösung eines Problems, die folgende Eigenschaften besitzen muss:
- Das Verfahren muss in einem endlichen Text eindeutig beschreibbar sein (Finitheit).
 - Jeder Schritt des Verfahrens muss auch tatsächlich ausführbar sein (Ausführbarkeit).
 - Das Verfahren muss nach endlich vielen Schritten abgeschlossen sein (Terminierung).
 - Das Verfahren muss effizient sein, es muss also mit den gegebenen Ressourcen auskommen.

Außerdem ggf.:

- Das Verfahren muss korrekt sein.
- Das Verfahren muss bei den gleichen Voraussetzungen das gleiche Ergebnis liefern (Determiniertheit).

Vergleiche Wikipedia „Algorithmus“ und 0-Intro.pdf Seite 4.

- (b) Parameter stellen die Schnittstelle einer Methode dar. Sie legen also fest, welche Informationen beim Aufruf einer Methode an diese übergeben werden können bzw. müssen. Argumente wiederum nennt man eben diese Informationen, die beim Methodenaufruf übergeben werden. Die zu übergebenden Argumente, die eine Methode erwartet, ergeben sich also aus den Parametern der Methode.

(c)

Elementarer Typ	Bit-Tiefe	Erläuterung
boolean	1 Bit	Speichert Wahrheitswerte (wahr und falsch).
char	16 Bit	Speichert einzelne Zeichen (bzw. deren numerischen Wert). Kann (intern) keine negativen Werte speichern.
byte	8 Bit	Speichern ganzzahlige Werte. Der maximale Wert ist von der jeweiligen Bit-Tiefe abhängig.
short	16 Bit	
int	32 Bit	
long	64 Bit	
float	32 Bit	Speichern Fließkommazahlen. Die maximale Anzahl Nachkommastellen variiert.
double	64 Bit	

- (d) Extreme Programming:
- Bei der Anwendungsentwicklung wird zu aller erst der Testcode geschrieben.

- b. Es soll nichts implementiert werden, das nicht gefordert ist.
- c. Es soll paarweise (von zwei Programmierern) programmiert werden: Der eine schreibt, der andere denkt mit. Die Rollen werden abgewechselt.

Vergleiche Java von Kopf bis Fuß, Seite 101 unten.

(e) Code überprüfen:

- `int zahl = Integer.parseInt („vier“);`
Laufzeitfehler, da die Methode `parseInt` nur als Zeichenkette notierte Ziffern umwandeln kann. (Der Compiler überprüft nur, ob ein String übergeben wird.)
Korrektur: `„4“` statt `„vier“`.
- `int x = 34.5;`
Compilerfehler, da `34.5` ein Fließkommawert ist. Korrektur: `double` statt `int`.
- `public static void main (string[] args) { ... }`
Compilerfehler, da der Name der Klasse `String` mit einem großen `S` beginnt.
Korrektur: `String[]` statt `string[]`.
- `If (x > 7) x++; else x = x - 1;`
OK.
- `public double quadrat (int x) return x * x;`
Compilerfehler, da die geschweiften Klammern fehlen. Korrektur: `...(int x) { return x * x; }`
- `float $dollar$;`
OK.

Aufgabe 2

Ausgabe.java

```

1:  /*
2:   * Klasse Ausgabe
3:   *
4:   * Gibt alle bei Programmaufruf uebergebenen Worte in umgekehrter
5:   * Reihenfolge aus.
6:   *
7:   */
8:
9:  public class Ausgabe {
10:
11:     public static void main (String[] args) {
12:
13:         // Array args von hinten nach vorne durchlaufen
14:         for (int i = args.length; i > 0; i--) {
15:             System.out.print (args[i] + " ");
16:         }
17:
18:     }
19:
20: }
```

Aufgabe 3

```

1:  public static void convertSec (int sec) {
2:     System.out.print(sec/3600 + ":" + (sec%3600)/60 + ":" + sec%60);
3: }
```

Aufgabe 4

```
1: public static double c2f (double c) {
2:     return 9d * c / 5d + 32d;
3: }
```

Hinweis: Der Operator / führt eine ganzzahlige Division durch, wenn Dividend und Divisor Integer sind. Dabei werden Nachkommastellen abgeschnitten. Da in diesem Beispiel die Variable c als Double deklariert ist, wird / jedoch keine ganzzahlige Division durchführen. Man könnte daher die Formel aus der Aufgabenstellung ohne Änderung übernehmen. Um dennoch dem Hinweis aus der Aufgabenstellung zu folgen, darauf zu achten keine Nachkommastellen zu „verlieren“, sind alle ganzen Zahlen in der Formel mit Hilfe des kleinen d explizit als Double ausgewiesen.

Aufgabe 5

- (a) Die Schleife läuft n mal. Bei jedem Schleifendurchlauf wird n zu res hinzuaddiert. Es wird also $n + n + n + \dots + n$ ausgerechnet, für n Summanden. Das Ergebnis ist n^2 .
- (b) Es wird der erste Teil der if-Anweisung ausgeführt. Danach ist $i = 3$ und $res = 4$. $res + 2 * i$ ergibt also **10** (Punkt vor Strich!).
- (c) Die äußere Schleife läuft x mal. Die innere Schleife läuft bei jedem Schleifendurchlauf der äußeren Schleife y mal. Die innere Schleife wird also $x * y$ mal durchlaufen.

Aufgabe 6

- (a)

vogel = adler;	OK
adler = vogel;	Compilerfehler
adler = (Adler) vogel;	Laufzeitfehler
adler = pinguin;	Compilerfehler
adler = (Adler) pinguin;	Compilerfehler

- (b)

```
if (vogel instanceof adler)
    adler = (Adler) vogel;
```

Ich bin nicht sicher, ob (b) richtig ist!

Aufgabe 7

- (a) Student.java

```
1: /*
2:  * Klasse Student
3:  *
4:  * Speichert Eigenschaften eines Studenten.
5:  *
6:  */
7:
8: public class Student {
9:
10:     // Eigenschaften
11:     protected String vorname;
12:     protected String nachname;
```

```

13:   protected int matrikelnummer;
14:   protected int semesterzahl;
15:   protected String studienfach;
16:
17:   // Getter und Setter Methoden
18:
19:   // Nachname
20:   public String getNachname () {
21:       return nachname;
22:   }
23:   public void setNachname (String neu) {
24:       nachname = neu;
25:   }
26:
27:   // Matrikelnummer
28:   public int getMatrikelnummer () {
29:       return matrikelnummer;
30:   }
31:   public void setMatrikelnummer (int neu) {
32:       if (neu >= 100000 && neu <= 300000)
33:           matrikelnummer = neu;
34:       else
35:           matrikelnummer = 0;
36:   }
37:
38:   // Main-Methode
39:
40:   public static void main (String[] args) {
41:   }
42:
43: }

```

```

(b) Student meinStudent = new Student ();
meinStudent.setVorname („Hans“);
meinStudent.setNachname („Wurst“);
meinStudent.setMatrikelnummer (123456);
meinStudent.setSemesterzahl (37);
meinStudent.setStudienfach („BWL“);

```