

Lösungsvorschlag PG2 – SS 2014

Von Tobias Graml & Ian Staples

Aufgabe 1 (Grundlagen)

- Worin unterscheidet sich die Programmiersprache Java von C? Nennen Sie drei Stichpunkte.
 - **Objektorientiert**
 - **Keinen expliziten Pointer-Typ**
 - **Auto. Speichermanagement (Garbage Collection)**
 - **Exception-Handling**
 - **Kein Überladen von Operatoren**
 - **Plattform-unabhängiger ByteCode**
-

Aufgabe 1 (Grundlagen)

- Nennen Sie vier Grunddatentypen von Java mit den zugehörigen Wertebereichen.

<u>Name</u>	<u>Länge (Bytes)</u>	<u>Wertebereich</u>	<u>Standardwert</u>
boolean	1	true, false	false
char	2	Alle Unicode-Zeichen	\u0000
byte	1	$-2^7 \dots 2^7 - 1$	0
short	2	$-2^{15} \dots 2^{15} - 1$	0
int	4	$-2^{31} \dots 2^{31} - 1$	0
long	8	$-2^{63} \dots 2^{63} - 1$	0
float	4	$\pm 3.40282347 \cdot 10^{38}$	0.0
double	8	$\pm 1.79769313486231570 \cdot 10^{308}$	0.0

Aufgabe 1 (Grundlagen)

- Worin unterscheidet sich UNICODE von ANSI? Nennen Sie einen Aspekt.
 - **Unicode: (z.B. UTF-8)**
 - 2 Byte = 65536 Zeichen (Version 1.0)
 - 128 ASCII-Zeichen + X
 - (Erweiterbar bis zu 4 Byte)
 - Heutiger Standard (Vor allem im Internet und bei Java [char])
 - **ANSI:**
 - 128 ASCII-Zeichen + 128 Weitere
 - => Speichert max. 256 Zeichen (1 Byte)
 - Veraltete Technik
-

Aufgabe 1 (Grundlagen)

- Rechnen Sie die Dualzahl 1110 in das Dezimalsystem um.

$$1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 = 14$$

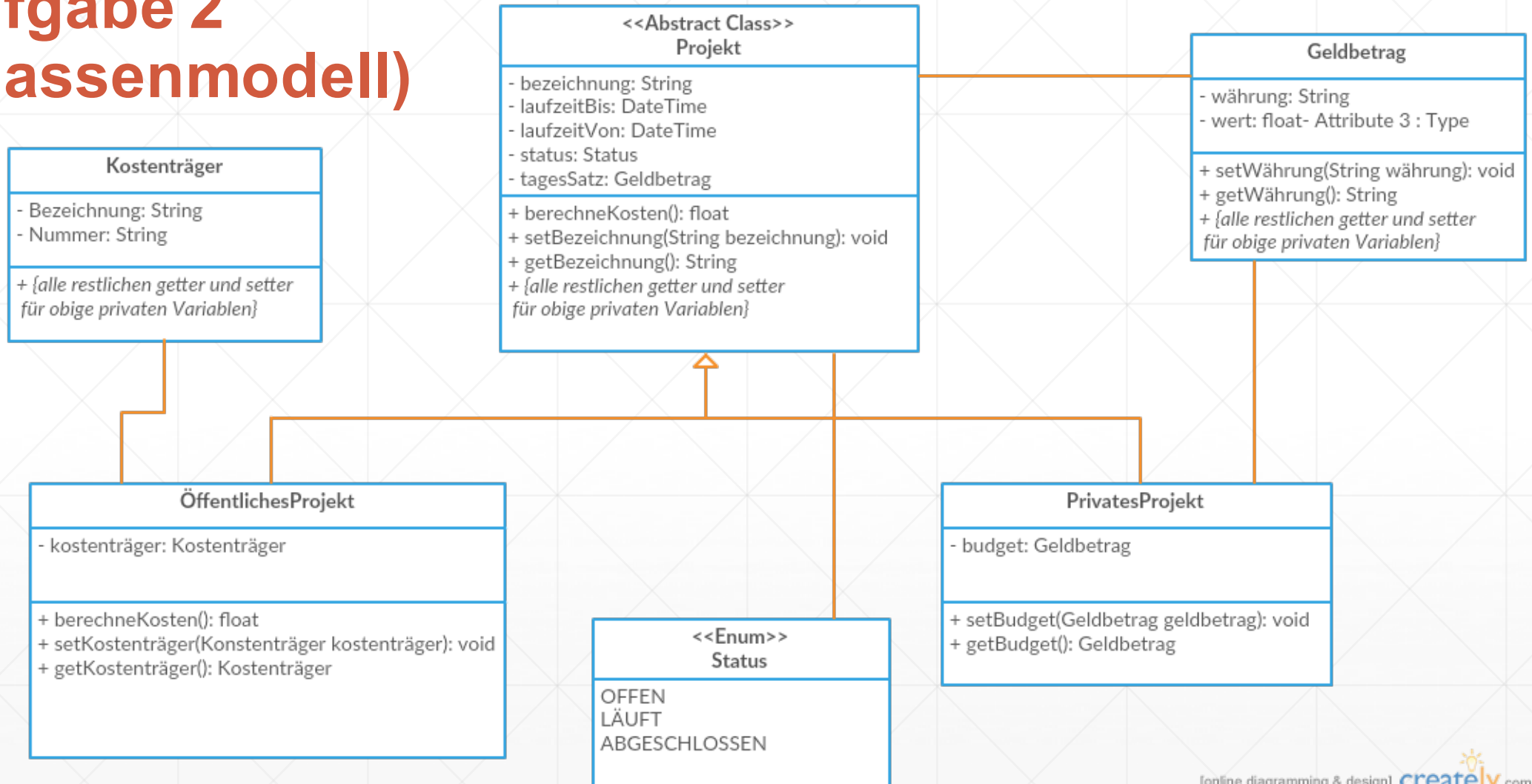
Aufgabe 1 (Grundlagen)

- Was ist die Aufgabe einer Betriebssystem-Umgebungsvariablen?

Sind Variablen des Betriebssystems, welche den Pfad zu Dateien/Programmen und Informationen über das Betriebssystem/User enthalten.

z.B. zum Setzen der JAVA_HOME Variable

Aufgabe 2 (Klassenmodell)



Aufgabe 2 (Klassenmodell, Programmierung)

- Programmieren Sie die Methoden, welche Sie für die Kostenberechnungen von privaten und öffentlichen Projekten vorgesehen haben.

```
12 public float aktuelleKosten() {
13     Date date = new Date();
14     //Überprüft, ob das Projekt bereits begonnen hat
15     if (laufzeitVon.getTime() > date.getTime()) {
16         return 0;
17     }
18     //Überprüft, ob das Projekt abgeschlossen ist
19     else if (laufzeitBis.getTime() < date.getTime()) {
20         long temp = laufzeitBis.getTime() - laufzeitVon.getTime();
21         //(1000[auf Sekunden] * 60[auf Minute * 60[auf Stunde] * 24 [auf Tag])
22         long diffInDays = temp / (1000 * 60 * 60 * 24);
23
24         return ((int) diffInDays) * Tagessatz;
25     }
26     //Projekt ist abgeschlossen, Berechnung der Projektdauer
27     else {
28         long temp = date.getTime() - laufzeitVon.getTime();
29         long diffInDays = temp / (1000 * 60 * 60 * 24);
30
31         return ((int) diffInDays) * Tagessatz;
32     }
33 }
34
```


Aufgabe 2 (Klassenmodell, Programmierung)

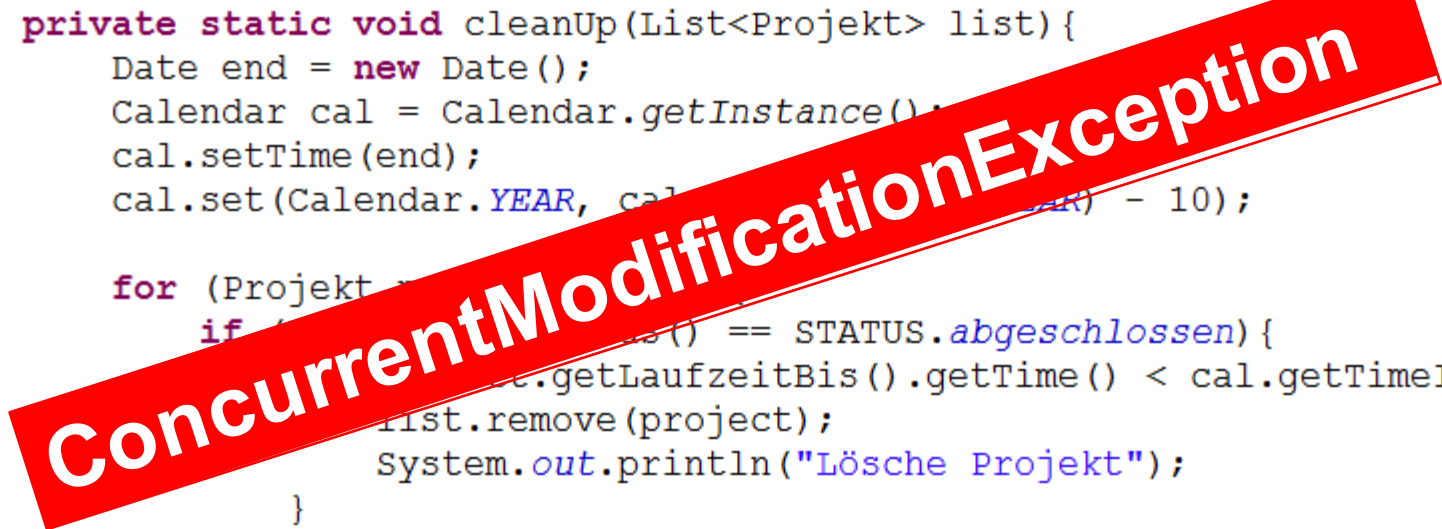
- In der Software gibt es eine Liste *list* von Projekten. Diese Liste soll absteigend nach den Projektkosten sortiert werden. Ergänzen Sie Ihre Klassen entsprechend und implementieren Sie die hierfür nötigen Änderungen bzw. Erweiterungen.

```
4 public abstract class Projekt implements Comparable<Projekt>{
5
6     @Override
7     public int compareTo(Pjekt o) { //Absteigende Sortierung
8         if (this.aktuelleKosten() > o.aktuelleKosten()){
9             return 1;
10        } else if (this.aktuelleKosten() < o.aktuelleKosten()){
11            return -1;
12        }
13        return 0;
14    }
15 }
```

Aufgabe 2 (Klassenmodell, Programmierung)

- In der Software gibt es weiterhin eine Liste *list* von Projekten. Programmieren Sie eine statische Methode mit dem Namen *cleanUp*, welche die Liste *list* als Parameter übergeben bekommt. Diese Methode löscht aus der übergebenen Liste alle Projekte, die den Status „abgeschlossen“ haben, wenn diese älter als 10 Jahre sind.

```
22 private static void cleanUp(List<Projekt> list){
23     Date end = new Date();
24     Calendar cal = Calendar.getInstance();
25     cal.setTime(end);
26     cal.set(Calendar.YEAR, cal.get(Calendar.YEAR) - 10);
27
28     for (Projekt projekt : list) {
29         if (projekt.getStatus() == STATUS.abgeschlossen) {
30             if (projekt.getLaufzeitBis().getTime() < cal.getTimeInMillis()) {
31                 list.remove(projekt);
32                 System.out.println("Lösche Projekt");
33             }
34         }
35     }
36 }
```



Aufgabe 2 (Klassenmodell, Programmierung)

- In der Software gibt es weiterhin eine Liste *list* von Projekten. Programmieren Sie eine statische Methode mit dem Namen *cleanUp*, welche die Liste *list* als Parameter übergeben bekommt. Diese Methode löscht aus der übergebenen Liste alle Projekte, die den Status „abgeschlossen“ haben, wenn diese älter als 10 Jahre sind.

```
/**
 * Lösung mit Iterator
 */

Iterator<Projekt> itProjekt = list.iterator();
while (itProjekt.hasNext()){
    Projekt projekt = itProjekt.next();
    if (projekt.getStatus() == STATUS.abgeschlossen){
        if (projekt.getLaufzeitBis().getTime() < cal.getTimeInMillis()){
            itProjekt.remove();
            System.out.println("Lösche Projekt");
        }
    }
}
```

Aufgabe 2 (Klassenmodell, Programmierung)

- In der Software gibt es weiterhin eine Liste *list* von Projekten. Programmieren Sie eine statische Methode mit dem Namen *cleanUp*, welche die Liste *list* als Parameter übergeben bekommt. Diese Methode löscht aus der übergebenen Liste alle Projekte, die den Status „abgeschlossen“ haben, wenn diese älter als 10 Jahre sind.

```
/**
 * Lösung mit removeAll und extra Liste
 */

List<Projekt> listToRemove = new ArrayList<>();
for (Projekt projekt : list){
    if (projekt.getStatus() == STATUS.abgeschlossen){
        if (projekt.getLaufzeitBis().getTime() < cal.getTimeInMillis()){
            listToRemove.add(projekt);
        }
    }
}
list.removeAll(listToRemove);
```

Aufgabe 3

- Beschreiben Sie die Hauptaufgabe der Methode *doIt* des folgenden Java-Programms in ca. 2 – 3 Sätzen. Welche Anforderung bzw. Funktionalität setzt die Methode um?

Zählt die Anzahl der Satzzeichen (aus dem Char-Array „p“) im übergebenen String „s“

```
1 public class WasMacheIch {
2
3     private static char[] p = { '.', '!', '?', ':' };
4
5     public static int doIt(String s) {
6         if (s == null){
7             return 0;
8         }
9         int x = 0;
10        char[] a = s.toCharArray();
11        for (char c : a) {
12            if (check(c)){
13                ++x;
14            }
15        }
16        return x;
17    }
18
19    private static boolean check(char a) {
20        for (char c : p) {
21            if (c == a) {
22                return true;
23            }
24        }
25        return false;
26    }
27 }
```

Aufgabe 3

- In dem folgenden Java-Programm sind zwei Fehler enthalten. Nennen Sie die Fehler und begründen Sie Ihre Wahl.

```
1 public class Fehlerhaft {  
2  
3     public static void main(String[] args) {  
4  
5         int[] x = new int[5];  
6  
7         for (int i = 5; i > 0 ; i--){  
8             System.out.println("Zahl: " + x[i]);  
9             if( x = 5 ){  
10                System.out.println("Zahl 5 ist enthalten");  
11            }  
12        }  
13    }  
14 }
```

IndexOutOfBoundsException
(Beginnt beim Index 6, zählen
beginnt bei Java bei 0)

Zuweisungsoperator statt
Vergleichsoperator und kein
Zugriff auf Arrayindex (x[i])

Aufgabe 4 (Threads)

```
~/  
public static void main(String[] args) {  
  
    for (int i = 0; i <= 1000; i++){  
        Jobs.addElement("Element " + i);  
    }  
    List<MyThread> jobList = new ArrayList<>();  
    for (int i = 0; i < 8; i++){  
        MyThread my = new MyThread();  
        jobList.add(my);  
        my.start();  
    }  
    for (MyThread job : jobList){  
        try {  
            job.join();  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
    System.out.println("Ende");  
}
```

```
public class MyThread extends Thread {  
  
    private synchronized String popElement(){  
        return Jobs.getElement();  
    }  
  
    public void run(){  
        String s = null;  
        while ((s = popElement()) != null){  
            System.out.println(s);  
        }  
    }  
}
```



**Viel Erfolg bei der
Prüfung**
