



Programmierung für Mathematik HS10

Übung 8

1 Aufgabe: Codeverständnis (Repetition)

1.1 Lernziele

1. Code verstehen können.
2. Fehler im Code finden und korrigieren können.

1.2 Aufgabenstellung

1. Überlegen Sie sich, was der Output bei folgendem Code wäre, wenn er so in einer main-Methode stehen würde. (Sie können dazu auch die Java-API verwenden)

```
1 String dog = "dog";
2 String cat = "cat";
3 cat.toUpperCase();
4 System.out.println(cat);
5 System.out.println(dog);
```

```
1 double x = 3.54;
2 int z = (int)x;
3 System.out.println(z/2);
```

```
1 double x = 1.25;
2 int z = 2;
3 System.out.println(z*x);
```

```
1 String cat = "CAT";
2 cat = cat.toLowerCase();
3 if (cat=="cat"){
4     System.out.println("is equal");
5 } else {
6     System.out.println("isn't equal");
7 }
```

```

1 for (int i=5; i>2; i--){
2     System.out.println(i+1);
3 }

```

2. Kompiliert folgender Code und wenn ja, welche Ausgabe wird produziert?

```

1 public class Animal {
2     public void print(){
3         System.out.println("it's an animal.");
4     }
5     public static void main(String[] args){
6         Animal a = new Dog();
7         a.print();
8     }
9 }

```

```

1 public class Dog extends Animal {
2     public void print(){
3         System.out.println("it's a dog.");
4     }
5 }

```

3. Welche Ausgabe wird von folgendem Code produziert?

```

1 public class TestDriver {
2     public static void main(String[] args){
3         System.out.println(A(2,1));
4     }
5     public static long A(long m, long n) {
6         if (m==0){
7             return n+1;
8         } else if (n==0) {
9             return A(m-1, 1);
10        } else {
11            return A(m-1, A(m, n-1));
12        }
13    }
14 }

```

4. Korrigieren Sie folgenden Code so, dass er korrekt kompiliert und auch zu Laufzeit keine Fehler auftreten.

```

1 public class TestDriver {
2     public static void main(String[] args){
3         int[] array = new int(3);
4         array[0] = 3;
5         array[2] = 1;
6         print(array)

```

```
7     }
8     public void print(int[] array) {
9         for (int i=0; i<=array.length; i++){
10            System.out.println(array[i]);
11        }
12    }
13 }
```

2 Aufgabe: Implementierung eines News-Dienstes

2.1 Lernziele

1. Problemstellung mit Hilfe von Interfaces sinnvoll implementieren können.
2. `ArrayList` verwenden können.

2.2 Aufgabenstellung

1. Bilden Sie folgenden Sachverhalt auf ein Interface, sowie auf Klassen, Attribute und Methoden ab:
Abonnenten (Subscriber), können sich bei verschiedenen News-Diensten (Newsletter) anmelden und erhalten dann stets Neuigkeiten zum Thema ihrer Wahl, sobald diese verfügbar werden (Abonnenten sollen erhaltene News einfach auf die Kommandozeile ausgeben). Jeder Dienst bietet News zu einem Thema (Topic) an. News-Dienste können nun wie folgt konfiguriert werden:
 - Dienste können sich auf ein sehr spezifisches Thema festlegen, also z.B. ein Dienst auf News zum Thema *Neuerscheinungen DVDs*, ein anderer zum Thema *Neuerscheinungen Bücher*.
 - Dienste können sich aber auch bei mehreren anderen Diensten anmelden (d.h. sie können selber auch als Abonnenten auftreten) und senden in dem Fall ihren eigenen Abonnenten die gesammelten News zu einem breiteren Themenkreis. Ein Beispiel wäre ein Dienst mit News zum Thema *Neuerscheinungen Medien*, der sich sowohl bei dem Dienst mit dem Thema *Neuerscheinungen DVDs*, als auch jenem mit dem Thema *Neuerscheinungen Bücher* anmeldet und sämtliche Neuigkeiten zu den beiden Themen an seine Abonnenten weiterleitet.
2. Schreiben Sie einen `TestDriver` um die Funktionalität Ihres Programmes zu testen.

3 Aufgabe: Nullstellen eines Polynoms

3.1 Lernziele

1. Selbstständig einen funktionierenden Algorithmus überlegen und implementieren können.

3.2 Aufgabenstellung

Ergänzen Sie Ihre Klasse `Polynomial` von Übung 4 mit der Methode

```
public double calculateRoot(double a, double b, double accuracy)
```

Diese Methode soll rekursiv eine Approximation einer Nullstelle des Polynoms p im Intervall $[a, b]$ berechnen, wobei angenommen werden darf, dass $p(a) \cdot p(b) \leq 0$, d.h. es gibt sicher eine Nullstelle in dem Intervall. Der Fehler der Approximation soll maximal "accuracy" betragen.

Tipp: Halbieren Sie das Intervall $[a, b]$, überprüfen Sie in welcher Hälfte die Nullstelle ist und verfahren Sie analog mit dieser Hälfte des Intervalls, bis die Intervalllänge "genügend klein" ist. Dann soll der Mittelpunkt des zuletzt berechneten Intervalls zurückgegeben werden.

4 Aufgabe: Euklid'scher Algorithmus

4.1 Lernziele

1. Iterativer Algorithmus in rekursiven Algorithmus umschreiben können.
2. Methoden überladen können.
3. Exceptions sinnvoll behandeln können.

4.2 Aufgabenstellung

1. Betrachten Sie folgende Methode:

```
1 public int euclidAlgorithm(int a, int b){
2     if (a==0){
3         return b;
4     }
5     while (b!=0){
6         if (a>b){
7             a = a-b;
8         } else {
9             b = b-a;
10        }
11    }
12    return a;
13 }
```

Diese Methode berechnet den grössten gemeinsamen Teiler von a und b . Schreiben Sie diese Methode in eine rekursive Methode um.

2. Schreiben Sie eine Klasse `Fraction`, welche einen Bruch darstellen soll. Der Bruch soll mit einem Konstruktor gesetzt werden können und automatisch in gekürzter Form abgespeichert werden. Falls der Benutzer für den Nenner 0 eingibt, soll eine Exception geworfen werden.
Die Klasse soll Methoden `add` und `multiply` zur Verfügung stellen, womit entweder ein anderer Bruch oder eine ganze Zahl addiert bzw. multipliziert werden können. Das Resultat soll direkt (in gekürzter Form) in den Instanzvariablen der Klasse abgespeichert werden.
3. Prüfen Sie Ihre Klasse in einem `TestDriver` und behandeln Sie allfällige Exceptions in einem `try-catch`-Block.