

Übungen zur Vorlesung
Objektorientierte Programmierung: Wintersemester 2022/2023

Nr. 3, Abgabe bis 14.11.2022

Aufgabe 3.1: Code Vein

6 Punkte

Coding Conventions sind Richtlinien für die Verwendung einer Sprache. Sie geben Standards für die Verwendung der Syntax vor und sorgen somit dafür, dass Programme leichter verstanden und gewartet werden können.

Im ILIAS finden Sie unter *Übungen* → *Code Conventions.pdf* eine Liste der wichtigsten Konventionen für diese Vorlesung.

Das Dokument wird im Laufe der Vorlesung um Konventionen für neue Sprachkonstrukte ergänzt.

Formatieren Sie den folgenden Programmcode entsprechend der Vorgaben. Achten Sie dabei darauf, dass auch alle Namen den Konventionen entsprechen. Deklarieren Sie außerdem alle Variablen und Parameter als **final**, deren Wert im Methodenverlauf nur einmal gesetzt und danach nicht mehr verändert wird.

```
1  double meine_berechnung(double basis, int exponent) {
2  double result = 1;
3  if(exponent<0) {
4  int anzahl=exponent* -1;
5  while( anzahl > 0) {
6  result = result/basis;
7  anzahl -= 1;
8  }
9  return result;
10 }else {
11 int Count = exponent;
12 while( Count > 0) {
13 result = result*basis;
14 Count-=1;
15 }
16 return result ;
17 }
18 }
19 void mein_test(){
20 boolean b = meine_berechnung(15,5) == 759375;
21 if(b) System.out.println("Test passed");
22 else System.out.println("Test failed");
23 }
```

Aufgabe 3.2: Please Mr. Postman

8 Punkte

Die folgende Tabelle enthält die aktuellen Portokosten für Briefsendungen der deutschen Post:

Format	Länge in mm	Breite in mm	Höhe in mm	Gewicht in g	Kosten in €
Postkarte	140 - 235	90 - 125	bis 2	bis 15	0,70
Standard	140 - 235	90 - 125	bis 5	bis 20	0,85
Kompakt	100 - 235	70 - 125	bis 10	bis 50	1,00
Groß	100 - 353	70 - 250	bis 20	bis 500	1,60
Maxi	100 - 353	70 - 250	bis 50	bis 1.000	2,75

a) Implementieren Sie zunächst die Hilfsfunktion

```
boolean isBetween(int value, int lower, int upper)
```

die **true** zurückgibt, wenn value zwischen lower und upper liegt
($lower \leq value \leq upper$).

b) Implementieren Sie die Methode

```
int postage(int length, int width, int height, int weight)
```

die das Porto aus den gegebenen Parametern berechnet und die Kosten in Cent zurückgibt.
Sollte das Versandstück in keine Kategorie fallen, soll -1 zurückgegeben werden.

c) Schreiben Sie eine ausreichende Anzahl an Testaufrufen für isBetween.

Hinweis: Sie können && (logisches und) verwenden, um multiple Bedingungen innerhalb eines if-blocks zu verknüpfen. Beispiel:

```
1 if (a > 5 && a <= 10) {  
2     ...  
3 }
```

Aufgabe 3.3: Meinten Sie Rekursion?

10 Punkte

Für die iterative Approximation von π kann eine spezielle Form der Madhava-Leibniz Reihe verwendet werden:

$$p_0 = \sqrt{12}$$
$$p_n = \sqrt{12} * \frac{(-\frac{1}{3})^n}{2n+1} + p_{n-1}$$

Hierbei gilt: $n \in \mathbb{N}^+$

Implementieren Sie Funktionen, die π mit Hilfe der Reihe approximieren und nach n Schritten abbrechen:

- a) `double piIterative (int n)`, die p_n iterativ mithilfe einer `while`-Schleife berechnet.
- b) `double piRecursive (int n)`, die p_n rekursiv berechnet.
- c) Testen Sie ihren Code mit drei verschiedenen n mit $0 \leq n \leq 20$
- d) Dokumentieren Sie ihre Lösung mit JavaDoc

4

4

1

1

Hinweis:

Um den Bruch $\frac{1}{3}$ zu verwenden, müssen Sie in Java `1.0/3.0` verwenden. Dies sorgt dafür, dass das Ergebnis vom Typ `double` ist und somit auch eine reelle Zahl sein kann.

Sie können die Funktionen `Math.sqrt` und `Math.pow` verwenden:

- `Math.sqrt (double a)` berechnet \sqrt{a}
- `Math.pow (double a, double b)` berechnet a^b