

Companion exercises
Objektorientierte Programmierung: Wintersemester 2021/2022

No. 3, due until 15.11.2021

Task 3.1: Code Vein

3 Points

Coding Conventions are conventions that are set for the use of a given language. They list all of the standards that are needed to make it easier to understand and maintain a programme. This is accomplished by imposing a set of rules on the usage of the syntax of the given language.

In our ILIAS-group, you can find a list of the most important rules for this course. You can find them by going to *Übungen* → *Code Conventions_en.pdf*.

The document will be updated over the course of the lecture when, for example, new features of the language are introduced.

Format the following code to be in line with the rules we set for this lecture. Take special care that all of the names of variables are in line with the rules. Furthermore, any variable that is initialized and never changed should be made **final**.

```
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 }
```

Task 3.2: Please Mr. Postman

4 Points

The following table contains the current cost of sending mail via Deutsche Post:

Form	Length in mm	Width in mm	Height in mm	Weight in g	Cost in €
Postcard	140 - 235	90 - 125	bis 2	bis 15	0.60
Standard	140 - 235	90 - 125	bis 5	bis 20	0.80
Compact	100 - 235	70 - 125	bis 10	bis 50	0.95
Big	100 - 353	70 - 250	bis 20	bis 500	1.55
Maxi	100 - 353	70 - 250	bis 50	bis 1.000	2.70

a) Implement your helper-function

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

that returns **true** if value is between lower and upper ($lower \leq value \leq upper$).

b) Implement a method

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

that computes the cost of mail with the given parameters in cents. Should a given object not fall within the criteria, your function should return -1.

c) Write a sufficient amount of test-calls for `isBetween`.

Attention: You can use `&&` (logical and) to compound multiple conditions within one `if`-block.
Example:

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

Task 3.3: Did you mean recursion?

5 Points

π can be iteratively approximated by using a special version of the Madhva-Leibniz sequence which is defined like this:

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

This only works if $n \in \mathbb{N}^+$

Implement functions that compute π using the above mentioned sequence and conclude after n steps (the function computes p_n):

- a) `double piIterative(int n)` that computes p_n using a `while`-loop.
- b) `double piRecursive(int n)` that computes p_n recursively.
- c) Test your solutions with three different n with $0 \leq n \leq 20$
- d) Document your implementation using JavaDoc

Attention:

To use $\frac{1}{3}$ you need to write `1.0/3.0` in Java. With this you ensure that your result will be of `double` type and a real number.

You may use functions `Math.sqrt` and `Math.pow`:

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