Prof. Dr. Christoph Bockisch
MSc Steffen Dick
Fachbereich Mathematik und Informatik
AG Programmiersprachen und -werkzeuge

Philipps Universität Marburg

Tutorial exercises
Objektorientierte Programmierung: Wintersemester 2021/2022
Nr. 6

Task 6.1: Tailmon

Look at the following recursive method to compute the faculty of a number:

```
1  int facultyRec(int n){
2    if(n==1 || n==0){
3      return 1;
4    }
5    return n * facultyRec(n-1);
6  }
```

a) Implement a method `int facultyTailRec(int n)` that only calls a recursive helper-method `int facultyTailH(int n, int f)` with correct values. Use `f` to save interim results.
b) Implement the recursive helper-method `int facultyTailH(int n, int f)` in such a way that it behaves exactly like `facultyRec`.
c) Test your implementation of `facultyTailRec` with at least 10 different calls and compare the results with the same calls to `facultyRec`.

Task 6.2: Zoe 101

You have learned how to create a class within this week's lecture.

a) Create a class called `Book` with the following fields:
   - `String` title
   - `String` author
   - `String` isbn
b) Create a constructor within the class `Book` in which all fields can be initialized.
c) Change all of your fields to be `final`. What do you think changes with your implementation?
d) Create some of your favourite books as objects of `Book`. Try to access all fields from outside of the class and print the values to the console.

Task 6.3: Double Trouble

As talked about in the lecture, floats and doubles are specially represented in programming. This can lead to not wanted behaviour within programmes. In this task, we want to investigate!

We're in double trouble tonight!

a) Implement a method `boolean doubleEqual(double a, double b)` that compares two numbers of type `double` using ==.
b) Call your method with values of `0.1 + 0.2` and `0.3` for `a` and `b` respectively. What result did you expect and what result did you get?
c) Write a method `boolean doubleEqualDelta(double a, double b, double d)` that returns `true` if $|a - b| < d$ is true.
d) Call your new method with the same values as last time for `a` and `b`. For `d` (delta) use `0.00000001`. What can you observe?