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. 9

Task 9.1: The curious occurrence of an index at String time

Implement a method `int countOccurences(String source, String search)` that returns the total count of `search` within `source`. Use `int indexOf(String str, int pos)` from the `String`-class within your implementation. This method takes the first occurrence of `str` from `pos` and returns its index.

Write at least one JUnit-test to test your implementation.

Task 9.2: Always check your references!

Implement a class `WholeNumber`. The class should have a private field of type `int`. Add a useful constructor to that class. Furthermore, implement methods `int getValue()` that returns the value of the private field and `void add(WholeNumber number)` that adds the value of `number` to the value of the object.

Now look at the following code-snippet:

```
1  @Test
2  public void test1(){
3    int a = 5;
4    int b = a;
5    b += 10;
6    Assert.assertEquals(a,b);
7  }
8
9  @Test
10 public void test2(){
11   WholeNumber a = new WholeNumber(5);
12   WholeNumber b = a;
13   b.add(new WholeNumber(10));
14   Assert.assertEquals(a,b);
15 }
```

What will be the result of `test1` and `test2`? Why do they succeed/fail? Give a reason for your answer!