

Hierbei handelt es sich weder um eine Veröffentlichung der Übungsleitung noch des Lehrstuhls für Software Engineering.

Dieses Dokument ist ein inoffizielles Übungsblatt für Studierende. Obwohl ich mich um Korrektheit bemühe, kann ich keine absolute Fehlerfreiheit garantieren. Wem ein Fehler auffällt bitte kurzen Hinweis per Mail an m.schwarz@tum.de.

Fehlersuche

Die folgenden Klassen enthalten 13 Fehler. Ziel ist es diese (natürlich ohne offene IDE) zu finden und kurz zu begründen, worin sie bestehen.

```
1 public class A {
2     public int aNumber = 42;
3     private static int i = 42;
4
5     public void evenMoreCode(B b) {
6         I i = (A) b;
7         this = b;
8     }
9
10    static void someMethod(I i) {
11        A a = (A) i;
12        a.evenMoreCode(new B());
13        C.evenMoreCode(new C());
14        C.someMethod(new C());
15    }
16
17    public D imRunningOuttaNames() {
18        new D();
19    }
20 }
21
22 public abstract class B extends A implements I {
23     private int aNumber = 42;
24     public double getSthElse() {
25         return -18.5;
26     }
27
28     protected D imRunningOuttaNames(int myPreciousInt) {
29         return null;
30     }
31 }
32
33 public class C extends B {
34     public int getSthElse() {
35         return i;
36     }
37
38     public int getWhatever() {
39         return aNumber;
40     }
41
42     protected D imRunningOuttaNames() {
43         return null;
44     }
45 }
46
47 public class D implements I {
48     public int getSth() {
49         return 17;
50     }
51
52     public double getSthElse() {
53         return 0;
54     }
55 }
56
57 public interface I {
58     public abstract int getSth();
59     public double getSthElse();
60     private void doSomething();
61
62     public void aMethod(){
63     }
64 }
```