

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 II

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

```
1 public interface J {
2     private void horst() {
3     }
4 }
5
6 public interface I extends J {
7     public void func();
8     public abstract int fancyName(int i);
9     protected void doSomething();
10 }
11
12 public class A implements I {
13     public int 88,someInt;
14     private int myInt = 13;
15
16     private void theCoolFunc(){
17         return 42;
18     }
19
20     public void func(){
21     }
22     protected abstract int fancyName();
23 }
24
25 public class B extends A {
26     private int someInt;
27
28     protected void func() {
29         theCoolFunc();
30     }
31
32     protected int singYouTheSongOfMyPeople() {
33         return 5.5;
34     }
35 }
36
37 public class C extends B implements I {
38     public static J whatever() {
39         singYouTheSongOfMyPeople();
40         return new J();
41     }
42
43     public int getTheNumber() {
44         int[33] = new int[33];
45         [int[31]] = -42;
46         return someInt;
47     }
48
49     public int getIt() {
50         return super.someInt;
51     }
52
53     public boolean returnSth(boolean b) {
54         if(b == true) {
55             return false;
56         }
57         if(!b) {
58             return true;
59         }
60     }
61 }
```