

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](mailto:m.schwarz@tum.de).

## Fehlersuche IV - Lösung

Die folgenden Klassen und Interfaces enthalten 12 Compiler-Fehler. Ziel ist es diese (natürlich ohne offene IDE) zu finden und kurz zu begründen, worin sie bestehen.

```
1 public interface I {
2     //Fehler: private unzulassig
3     private void name();
4     public void theFunc();
5
6     public int getX();
7 }
8
9 public interface J extends I {
10     public int getX();
11 }
12
13 public class A {
14     //Fehler: Abstrakte Methode in nicht abstrakter Klasse
15     public void abstract someMethod();
16
17     public int x;
18
19     public int getX()
20     {
21         return x;
22     }
23 }
24
25 public class B extends A {
26     private int x;
27
28     //Fehler: Weniger allgemeiner Zugriffsmodifikator nicht zulaessig
29     private int getX()
30     {
31         return 19;
32     }
33
34     private int getY()
35     {
36         return -19;
37     }
38 }
39
40 public class C extends B implements J {
41     //Fehler: Methode aus I ist nicht implementiert
42     //J.getX() schon (geerbt aus X)
43
44
45     public void theFunc() {
46         //Fehler: this kann ich nie etwas zuweisen
47         this = new C();
48     }
49
50     public void doSth() {
51         //Fehler: x ist verschattet -> beide Seiten falsch
52         this.x = super.x;
53     }
54
55     public void doSthElse() {
56         //Kein Fehler !!!
57         ((A)this).x = 33;
58     }
59
60     public void tuWasAnderes() {
61         this.getX();
62
63         //Fehler: Auf getY() kann nicht zugegriffen werden
64         this.getY();
65     }
66 }
```