

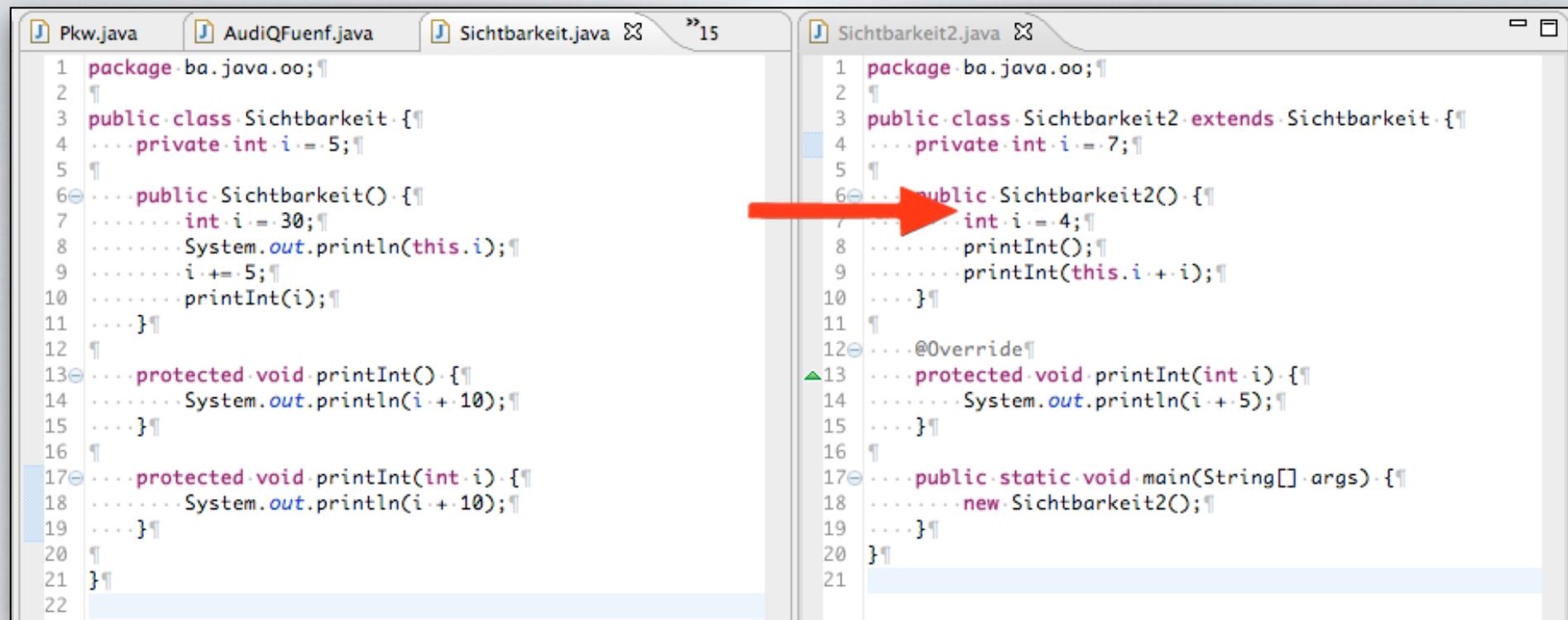


# PROGRAMMIEREN IN **Java**

TINF15 - Sommersemester 2016  
von Johannes Unterstein - [unterstein@me.com](mailto:unterstein@me.com)

```
Pkw.java
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt();
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

```
Sichtbarkeit2.java
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12     @Override
13     protected void printInt(int i) {
14         System.out.println(i + 5);
15     }
16
17     public static void main(String[] args) {
18         new Sichtbarkeit2();
19     }
20 }
21
```



```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt(i);
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20 }
```

The screenshot displays two Java code editors side-by-side in a development environment.

**Left Editor (Sichtbarkeit.java):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt(i);
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

**Right Editor (Sichtbarkeit2.java):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20 }
```

A red arrow points to the first line of the constructor in the Sichtbarkeit.java code.

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt(i);
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20 }
```

**Sichtbarkeit.java**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i); Ausgabe "5"
9         i += 5;
10        printInt(); this.i = Attribute des Objektes
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

**Sichtbarkeit2.java**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12     @Override
13     protected void printInt(int i) {
14         System.out.println(i + 5);
15     }
16
17     public static void main(String[] args) {
18         new Sichtbarkeit2();
19     }
20 }
```

The image shows a Java development environment with two code editors side-by-side.

**Left Editor (Sichtbarkeit.java):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt(i);           i der Methode wird um 5
11    }                         inkrementiert
12
13 protected void printInt() {
14     System.out.println(i + 10);
15 }
16
17 protected void printInt(int i) {
18     System.out.println(i + 10);
19 }
20
21 }
```

**Right Editor (Sichtbarkeit2.java):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12 @Override
13 protected void printInt(int i) {
14     System.out.println(i + 5);
15 }
16
17 public static void main(String[] args) {
18     new Sichtbarkeit2();
19 }
20 }
```

A red arrow points from the explanatory text "i der Methode wird um 5 inkrementiert" in the left editor to the line of code "i += 5;" in the Sichtbarkeit.java file.

The screenshot shows a Java development environment with two code editors side-by-side.

**Left Editor (Sichtbarkeit.java):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10    printInt(i); Aufruf mit i der Methode (35)
11 }
12
13 protected void printInt() {
14     System.out.println(i + 10);
15 }
16
17 protected void printInt(int i) {
18     System.out.println(i + 10);
19 }
20
21 }
```

**Right Editor (Sichtbarkeit2.java):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12 @Override
13 protected void printInt(int i) {
14     System.out.println(i + 5);
15 }
16
17 public static void main(String[] args) {
18     new Sichtbarkeit2();
19 }
20 }
```

A red arrow points to the line of code in Sichtbarkeit.java where `i` is used as a parameter in the call to `printInt(i)`, specifically the line **Aufruf mit i der Methode (35)**.

The screenshot shows two Java code editors side-by-side. The left editor contains the code for the `Sichtbarkeit` class, and the right editor contains the code for the `Sichtbarkeit2` class.

**Pkw.java (Left Editor):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt(i);      Es wird printInt(i) der
11    }                      konkretesten Klasse aufgerufen
12
13    protected void printInt() {
14        System.out.println(i + 10);
15    }
16
17    protected void printInt(int i) {
18        System.out.println(i + 10);
19    }
20
21 }
```

**Sichtbarkeit2.java (Right Editor):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt() {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20 }
```

A red arrow points from the annotation `@Override` in the `printInt()` method of `Sichtbarkeit2` to the `printInt()` method in the `Sichtbarkeit` class, illustrating that the `printInt()` method in `Sichtbarkeit2` overrides the one in `Sichtbarkeit`.

The screenshot displays two Java code editors side-by-side:

**Pkw.java:**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10    printInt(i);      Ausgabe 35+5 -> "40"
11 }
12
13 protected void printInt() {
14     System.out.println(i + 10);
15 }
16
17 protected void printInt(int i) {
18     System.out.println(i + 10);
19 }
20
21 }
```

**Sichtbarkeit2.java:**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20 }
```

A red arrow points from the line `protected void printInt(int i) {` in the Sichtbarkeit2.java editor to the line `protected void printInt() {` in the Pkw.java editor.

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt(i);
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20
21 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt();
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20 }
```

The screenshot shows two Java code editors side-by-side. The left editor contains the file `Pkw.java` with the following code:

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt();
11    }
12
13    protected void printInt() {
14        System.out.println(i + 10);
15    }
16
17    protected void printInt(int i) {
18        System.out.println(i + 10);
19    }
20
21 }
```

The right editor contains the file `Sichtbarkeit2.java` with the following code:

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20 }
```

A red arrow points to the `protected void printInt()` method in `Pkw.java`. Below the code in the left editor, there is a text overlay that reads "Ausgabe Attribute i (5) +10" and "=> "15".

The image shows a Java development environment with two code editors side-by-side.

**Sichtbarkeit.java (Left Editor):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5; Aufruf mit Attribute von Sichtbarkeit2 (i=7) plus i aus Methode (4) -> printInt(11)
10        printInt(i);
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

**Sichtbarkeit2.java (Right Editor):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12     @Override
13     protected void printInt(int i) {
14         System.out.println(i + 5);
15     }
16
17     public static void main(String[] args) {
18         new Sichtbarkeit2();
19     }
20 }
```

A red arrow points from the annotation in the Sichtbarkeit.java code to the line `printInt(this.i + i);` in the Sichtbarkeit2.java code.

The image shows two Java code editors side-by-side. The left editor contains the code for class `Sichtbarkeit`, and the right editor contains the code for class `Sichtbarkeit2`. A red arrow points from the `protected void printInt()` method in `Sichtbarkeit` to its overridden version in `Sichtbarkeit2`.

**Pkw.java (Left Editor):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt();
11    }
12
13    protected void printInt() {
14        System.out.println(i + 10);
15    }
16
17    protected void printInt(int i) {
18        System.out.println(i + 10);
19    }
20
21 }
```

**Ausgabe von 11 + 5 -> "16"**

**Sichtbarkeit2.java (Right Editor):**

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt();
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12     @Override
13     protected void printInt(int i) {
14         System.out.println(i + 5);
15     }
16
17     public static void main(String[] args) {
18         new Sichtbarkeit2();
19     }
20 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit {
4     private int i = 5;
5
6     public Sichtbarkeit() {
7         int i = 30;
8         System.out.println(this.i);
9         i += 5;
10        printInt();
11    }
12
13     protected void printInt() {
14         System.out.println(i + 10);
15     }
16
17     protected void printInt(int i) {
18         System.out.println(i + 10);
19     }
20
21 }
```

```
1 package ba.java.oo;
2
3 public class Sichtbarkeit2 extends Sichtbarkeit {
4     private int i = 7;
5
6     public Sichtbarkeit2() {
7         int i = 4;
8         printInt();
9         printInt(this.i + i);
10    }
11
12    @Override
13    protected void printInt(int i) {
14        System.out.println(i + 5);
15    }
16
17    public static void main(String[] args) {
18        new Sichtbarkeit2();
19    }
20
21 }
```