

Script generated by TTT

Title: groh: profile1 (10.06.2016)

Date: Fri Jun 10 13:29:20 CEST 2016

Duration: 94:29 min

Pages: 76

Access Modifiers, Packages

- Access Modifier für Methoden und Attribute:

| | Class | Package | Subclasses | World |
|-------------|-------|---------|------------|-------|
| public | ✓ | ✓ | ✓ | ✓ |
| protected | ✓ | ✓ | ✓ | |
| no modifier | ✓ | ✓ | | |
| private | ✓ | | | |

- Packages:
 - Kapseln (hierarchisch organisiert) Mengen von Klassen und Interfaces
 - Deklaration: package *nameOfPackage*;
 - Beispiele: `java.math`, `java.lang`, `java.net`, `de.tum.in`

Access Modifiers, Packages

- Access Modifier für Methoden und Attribute:

| | Class | Package | Subclasses | World |
|-------------|-------|---------|------------|-------|
| public | ✓ | ✓ | ✓ | ✓ |
| protected | ✓ | ✓ | ✓ | |
| no modifier | ✓ | ✓ | | |
| private | ✓ | | | |

Access Modifiers, Packages

- Access Modifier für Methoden und Attribute:

| | Class | Package | Subclasses | World |
|-------------|-------|---------|------------|-------|
| public | ✓ | ✓ | ✓ | ✓ |
| protected | ✓ | ✓ | ✓ | |
| no modifier | ✓ | ✓ | | |
| private | ✓ | | | |

- Packages:

- Kapseln (hierarchisch organisiert) Mengen von Klassen und Interfaces
- Deklaration: package *nameOfPackage*;
- Beispiele: `java.math`, `java.lang`, `java.net`, `de.tum.in`

- Packages:

- Kapseln (hierarchisch organisiert) Mengen von Klassen und Interfaces
- Deklaration: package *nameOfPackage*;
- Beispiele: `java.math`, `java.lang`, `java.net`, `de.tum.in`

Access Modifiers: static und final

- static:

- Methode oder Attribut gehört zur Klasse und nicht zum Objekt
(Attribut: existiert nur einmal, ist für alle Objekte dasselbe)
„Klassenmethode, Klassenattribut“ <---> „Instanzenmethode, Instanzenattribut“

- final:

- für Attribute: können nicht mehr geändert werden (Konstanten)
- für Methoden: können nicht overridden oder hidden werden (kommt gleich)
- für Klassen: Es können keine Unterklassen davon abgeleitet werden.

```
final class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
  
    static void methodOne() { /* ... */ }  
    final void methodTwo() { /* ... */ }  
    static final void methodThree() { /* ... */ }  
    void methodFour() { /* ... */ }  
  
}
```

Access Modifiers: static und final

- static:

- Methode oder Attribut gehört zur Klasse und nicht zum Objekt
(Attribut: existiert nur einmal, ist für alle Objekte dasselbe)
„Klassenmethode, Klassenattribut“ <---> „Instanzenmethode, Instanzenattribut“

- final:

- für Attribute: können nicht mehr geändert werden (Konstanten)
- für Methoden: können nicht overridden oder hidden werden (kommt gleich)
- für Klassen: Es können keine Unterklassen davon abgeleitet werden.

```
final class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
  
    static void methodOne() { /* ... */ }  
    final void methodTwo() { /* ... */ }  
    static final void methodThree() { /* ... */ }  
    void methodFour() { /* ... */ }  
  
}
```

Access Modifiers: static und final

- static:

- Methode oder Attribut gehört zur Klasse und nicht zum Objekt
(Attribut: existiert nur einmal, ist für alle Objekte dasselbe)
„Klassenmethode, Klassenattribut“ <---> „Instanzenmethode, Instanzenattribut“

- final:

- für Attribute: können nicht mehr geändert werden (Konstanten)
- für Methoden: können nicht overridden oder hidden werden (kommt gleich)
- für Klassen: Es können keine Unterklassen davon abgeleitet werden.

```
final class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
  
    static void methodOne() { /* ... */ }  
    final void methodTwo() { /* ... */ }  
    static final void methodThree() { /* ... */ }  
    void methodFour() { /* ... */ }  
  
}
```

Access Modifiers: static und final

- static:

- Methode oder Attribut gehört zur Klasse und nicht zum Objekt
(Attribut: existiert nur einmal, ist für alle Objekte dasselbe)
„Klassenmethode, Klassenattribut“ <---> „Instanzenmethode, Instanzenattribut“

- final:

- für Attribute: können nicht mehr geändert werden (Konstanten)
- für Methoden: können nicht overridden oder hidden werden (kommt gleich)
- für Klassen: Es können keine Unterklassen davon abgeleitet werden.

```
final class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;   
  
    static void methodOne() { /* ... */ }  
    final void methodTwo() { /* ... */ }  
    static final void methodThree() { /* ... */ }  
    void methodFour() { /* ... */ }  
  
}
```

Access Modifiers: static und final

```
public class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
    public int instVar;  
  
    public MyClass(int iv, int const) {  
        instVar = iv;  
        constantMayBeDifferentForEachInstance = const;  
    } // constructor  
  
    public static void methodOne() { /* ... */ }  
    public void methodTwo() { /* ... */ }  
}
```

```
MyClass m1 = new MyClass(20, 11);  
MyClass m2 = new MyClass(30, 9);  
MyClass.CONSTANT_SAME_FOR_ALL_INSTANCES = 3333; //ERROR  
MyClass.sameForAllInstances = 99;  
System.out.println(m1.sameForAllInstances); // 99  
System.out.println(m2.sameForAllInstances); // 99  
System.out.println(MyClass.sameForAllInstances); // 99  
System.out.println(m1.instVar); // 20  
System.out.println(m2.instVar); // 30  
m1.instVar = 77;  
MyClass.methodOne();  
m1.methodTwo();  
MyClass.methodTwo(); //ERROR
```

Muss im Constructor gesetzt werden (oder mit Initializer direkt bei Deklaration (bspw. final int c = 0;))

137

Access Modifiers: static und final

```
public class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
    public int instVar;  
  
    public MyClass(int iv, int const) {  
        instVar = iv;  
        constantMayBeDifferentForEachInstance = const;  
    } // constructor  
  
    public static void methodOne() { /* ... */ }  
    public void methodTwo() { /* ... */ }  
}
```

```
MyClass m1 = new MyClass(20, 11);  
MyClass m2 = new MyClass(30, 9);  
MyClass.CONSTANT_SAME_FOR_ALL_INSTANCES = 3333; //ERROR  
MyClass.sameForAllInstances = 99;  
System.out.println(m1.sameForAllInstances); // 99  
System.out.println(m2.sameForAllInstances); // 99  
System.out.println(MyClass.sameForAllInstances); // 99  
System.out.println(m1.instVar); // 20  
System.out.println(m2.instVar); // 30  
m1.instVar = 77;  
MyClass.methodOne();  
m1.methodTwo();  
MyClass.methodTwo(); //ERROR
```

Muss im Constructor gesetzt werden (oder mit Initializer direkt bei Deklaration (bspw. final int c = 0;))

137

Access Modifiers: static und final

```
public class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
    public int instVar;  
  
    public MyClass(int iv, int const) {  
        instVar = iv;  
        constantMayBeDifferentForEachInstance = const;  
    } // constructor  
  
    public static void methodOne() { /* ... */ }  
    public void methodTwo() { /* ... */ }  
}
```

```
MyClass m1 = new MyClass(20, 11);  
MyClass m2 = new MyClass(30, 9);  
MyClass.CONSTANT_SAME_FOR_ALL_INSTANCES = 3333; //ERROR  
MyClass.sameForAllInstances = 99;  
System.out.println(m1.sameForAllInstances); // 99  
System.out.println(m2.sameForAllInstances); // 99  
System.out.println(MyClass.sameForAllInstances); // 99  
System.out.println(m1.instVar); // 20  
System.out.println(m2.instVar); // 30  
m1.instVar = 77;  
MyClass.methodOne();  
m1.methodTwo();  
MyClass.methodTwo(); //ERROR
```

Muss im Constructor gesetzt werden (oder mit Initializer direkt bei Deklaration (bspw. final int c = 0;))

137

Access Modifiers: static und final

```
public class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
    public int instVar;  
  
    public MyClass(int iv, int const) {  
        instVar = iv;  
        constantMayBeDifferentForEachInstance = const;  
    } // constructor  
  
    public static void methodOne() { /* ... */ }  
    public void methodTwo() { /* ... */ }  
}
```

```
MyClass m1 = new MyClass(20, 11);  
MyClass m2 = new MyClass(30, 9);  
MyClass.CONSTANT_SAME_FOR_ALL_INSTANCES = 3333; //ERROR  
MyClass.sameForAllInstances = 99;  
System.out.println(m1.sameForAllInstances); // 99  
System.out.println(m2.sameForAllInstances); // 99  
System.out.println(MyClass.sameForAllInstances); // 99  
System.out.println(m1.instVar); // 20  
System.out.println(m2.instVar); // 30  
m1.instVar = 77;  
MyClass.methodOne();  
m1.methodTwo();  
MyClass.methodTwo(); //ERROR
```

Muss im Constructor gesetzt werden (oder mit Initializer direkt bei Deklaration (bspw. final int c = 0;))

137

Access Modifiers: static und final

```
public class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
    public int instVar;  
  
    public MyClass(int iv, int const) {  
        instVar = iv;  
        constantMayBeDifferentForEachInstance = const;  
    } // constructor  
  
    public static void methodOne() { /* ... */ }  
    public void methodTwo() { /* ... */ }  
}
```

```
MyClass m1 = new MyClass(20, 11);  
MyClass m2 = new MyClass(30, 9);  
MyClass.CONSTANT_SAME_FOR_ALL_INSTANCES = 3333; //ERROR  
MyClass.sameForAllInstances = 99;  
System.out.println(m1.sameForAllInstances); // 99  
System.out.println(m2.sameForAllInstances); // 99  
System.out.println(MyClass.sameForAllInstances); // 99  
System.out.println(m1.instVar); // 20  
System.out.println(m2.instVar); // 30  
m1.instVar = 77;  
MyClass.methodOne();  
m1.methodTwo();  
MyClass.methodTwo(); //ERROR
```

Muss im Constructor gesetzt werden (oder mit Initializer direkt bei Deklaration (bspw. final int c = 0;))

137

Access Modifiers: static und final

```
public class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
    public int instVar;  
  
    public MyClass(int iv, int const) {  
        instVar = iv;  
        constantMayBeDifferentForEachInstance = const;  
    } // constructor  
  
    public static void methodOne() { /* ... */ }  
    public void methodTwo() { /* ... */ }  
}
```

Muss im Constructor gesetzt werden (oder mit Initializer direkt bei Deklaration (bspw. final int c = 0;))

```
MyClass m1 = new MyClass(20, 11);  
MyClass m2 = new MyClass(30, 9);  
MyClass.CONSTANT_SAME_FOR_ALL_INSTANCES = 3333; //ERROR  
MyClass.sameForAllInstances = 99;  
System.out.println(m1.sameForAllInstances); // 99  
System.out.println(m2.sameForAllInstances); // 99  
System.out.println(MyClass.sameForAllInstances); // 99  
System.out.println(m1.instVar); // 20  
System.out.println(m2.instVar); // 30  
m1.instVar = 77;  
MyClass.methodOne();  
m1.methodTwo();  
MyClass.methodTwo(); //ERROR
```

137

Access Modifiers: static und final

```
public class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
    public int instVar;  
  
    public MyClass(int iv, int const) {  
        instVar = iv;  
        constantMayBeDifferentForEachInstance = const;  
    } // constructor  
  
    public static void methodOne() { /* ... */ }  
    public void methodTwo() { /* ... */ }  
}
```

Muss im Constructor gesetzt werden (oder mit Initializer direkt bei Deklaration (bspw. final int c = 0;))

137

Access Modifiers: static und final

```
public class MyClass {  
    static int sameForAllInstances = 3;  
    final int constantMayBeDifferentForEachInstance;  
    static final int CONSTANT_SAME_FOR_ALL_INSTANCES = 7;  
    public int instVar;  
  
    public MyClass(int iv, int const) {  
        instVar = iv;  
        constantMayBeDifferentForEachInstance = const;  
    } // constructor  
  
    public static void methodOne() { /* ... */ }  
    public void methodTwo() { /* ... */ }  
}
```

Muss im Constructor gesetzt werden (oder mit Initializer direkt bei Deklaration (bspw. final int c = 0;))

```
MyClass m1 = new MyClass(20, 11);  
MyClass m2 = new MyClass(30, 9);  
MyClass.CONSTANT_SAME_FOR_ALL_INSTANCES = 3333; //ERROR  
MyClass.sameForAllInstances = 99;  
System.out.println(m1.sameForAllInstances); // 99  
System.out.println(m2.sameForAllInstances); // 99  
System.out.println(MyClass.sameForAllInstances); // 99  
System.out.println(m1.instVar); // 20  
System.out.println(m2.instVar); // 30  
m1.instVar = 77;  
MyClass.methodOne();  
m1.methodTwo();  
MyClass.methodTwo(); //ERROR
```

137

-3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee();
9         LittleBee willi = new LittleBee();
10        maja.fly();
11        maja.sting();
12        willi.snooze();
13        AngryHornet evil;
14        evil = new AngryHornet();
15        evil.sting();
16
17        ICanSting someStinger;
18        someStinger = evil;
19        someStinger.sting();
20
21        Flower flower1 = new Flower();
22        Flower flower2 = new Flower();
23
24        maja.collectPollen(flower1, 20.0d);
25
26    }
27
28 }
```

-4Wzw/src/zue4Wzw/Demo.java - Eclipse

```
1 package zue4Wzw;
2
3 public class Demo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int result = 0;
8         Demo demo = new Demo();
9         result = demo.minimum(5, 10*2);
10        System.out.println("das ergebnis ist: " + String.valueOf(result));
11        long result22 = demo.faculty(-3);
12        System.out.println("das ergebnis ist: " + String.valueOf(result22));
13        double result222 = demo.powerWithWhile(2.0d, 3);
14        System.out.println("das ergebnis ist: " + String.valueOf(result222));
15        double result223 = demo.exp(2.0d);
16        System.out.println("das ergebnis ist: " + String.valueOf(result223));
17        double[] theArray = new double[3];
18        theArray[0] = 1.0d;
19        theArray[1] = 2.0d;
20        theArray[2] = 3.0d;
21        double result3 = demo.expectation(theArray);
22        System.out.println("das ergebnis ist: " + String.valueOf(result3));
23        double result4 = demo.variance(theArray);
24        System.out.println("das ergebnis ist: " + String.valueOf(result4));
25        double[][] matrix = new double[3][3];
26        System.out.println("das ergebnis ist: " + String.valueOf(matrix[2][2]));
27
28    }
29
30    public double variance(double a[]){
31        double result = 0.0d;
32        double oneOverNminusOne = 0.0d;
33        if(a.length >2)
34            oneOverNminusOne = 1.0d / ((double)(a.length) - 1);
35        else
36            System.out.println("array is not long enough");
37        double expectation = expectation(a);
38        for(int i=0; i<a.length; i++){
39            result = result + (a[i] - expectation) * (a[i] - expectation);
40        }
41        result = oneOverNminusOne * result;
42        return result;
43    }
44
45    public double expectation(double[] a){
46        double result = 0.0d;
47        double oneOverN = 0.0d;
48        if(a.length >= 1)
49            oneoverN = 1.0d / a.length;
50        else
51            System.out.println(" das array ist leer");
52        for(int i=0; i < a.length; i++){
53            result = result + a[i];
54        }
55        return result * oneOverN;
56    }
57
58    public int minimum(int a, int b){
59        int result;
```

-4Wzw/src/zue4Wzw/Demo.java - Eclipse

```
1 package zue4Wzw;
2
3 public class Demo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int result = 0;
8         Demo demo = new Demo();
9         /*result = demo.minimum(5, 10*2);
10        System.out.println("das ergebnis ist: " + String.valueOf(result));
11        long result22 = demo.faculty(-3);
12        System.out.println("das ergebnis ist: " + String.valueOf(result22));
13        double result222 = demo.powerWithWhile(2.0d, 3);
14        System.out.println("das ergebnis ist: " + String.valueOf(result222));
15        double result223 = demo.exp(2.0d);
16        System.out.println("das ergebnis ist: " + String.valueOf(result223));
17        double[] theArray = new double[3];
18        theArray[0] = 1.0d;
19        theArray[1] = 2.0d;
20        theArray[2] = 3.0d;
21        double result3 = demo.expectation(theArray);
22        System.out.println("das ergebnis ist: " + String.valueOf(result3));
23        double result4 = demo.variance(theArray);
24        System.out.println("das ergebnis ist: " + String.valueOf(result4));
25        double[][] matrix = new double[3][3];
26        System.out.println("das ergebnis ist: " + String.valueOf(matrix[2][2]));
27
28    }
29
30    public double variance(double a[]){
31        double result = 0.0d;
32        double oneOverNminusOne = 0.0d;
33        if(a.length >2)
34            oneOverNminusOne = 1.0d / ((double)(a.length) - 1);
35        else
36            System.out.println("array is not long enough");
37        double expectation = expectation(a);
38        for(int i=0; i<a.length; i++){
39            result = result + (a[i] - expectation) * (a[i] - expectation);
40        }
41        result = oneOverNminusOne * result;
42        return result;
43    }
44
45    public double expectation(double[] a){
46        double result = 0.0d;
47        double oneOverN = 0.0d;
48        if(a.length >= 1)
49            oneoverN = 1.0d / a.length;
50        else
51            System.out.println(" das array ist leer");
52        for(int i=0; i < a.length; i++){
53            result = result + a[i];
54        }
55        return result * oneOverN;
56    }
57
58    public int minimum(int a, int b){
59        int result;
```

-4Wzw/src/zue4Wzw/Demo.java - Eclipse

```
1 package zue4Wzw;
2
3 public class Demo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int result = 0;
8         Demo demo = new Demo();
9         /*result = demo.minimum(5, 10*2);
10        System.out.println("das ergebnis ist: " + String.valueOf(result));
11        long result22 = demo.faculty(-3);
12        System.out.println("das ergebnis ist: " + String.valueOf(result22));
13        double result222 = demo.powerWithWhile(2.0d, 3);
14        System.out.println("das ergebnis ist: " + String.valueOf(result222));
15        double result223 = demo.exp(2.0d);
16        System.out.println("das ergebnis ist: " + String.valueOf(result223));
17        double[] theArray = new double[3];
18        theArray[0] = 1.0d;
19        theArray[1] = 2.0d;
20        theArray[2] = 3.0d;
21        double result3 = demo.expectation(theArray);
22        System.out.println("das ergebnis ist: " + String.valueOf(result3));
23        double result4 = demo.variance(theArray);
24        System.out.println("das ergebnis ist: " + String.valueOf(result4));
25        double[][] matrix = new double[3][3];
26        System.out.println("das ergebnis ist: " + String.valueOf(matrix[2][2]));
27
28    }
29
30    public double variance(double a[]){
31        double result = 0.0d;
32        double oneOverNminusOne = 0.0d;
33        if(a.length >2)
34            oneOverNminusOne = 1.0d / ((double)(a.length) - 1);
35        else
36            System.out.println("array is not long enough");
37        double expectation = expectation(a);
38        for(int i=0; i<a.length; i++){
39            result = result + (a[i] - expectation) * (a[i] - expectation);
40        }
41        result = oneOverNminusOne * result;
42        return result;
43    }
44
45    public double expectation(double[] a){
46        double result = 0.0d;
47        double oneOverN = 0.0d;
48        if(a.length >= 1)
49            oneoverN = 1.0d / a.length;
50        else
51            System.out.println(" das array ist leer");
52        for(int i=0; i < a.length; i++){
53            result = result + a[i];
54        }
55        return result * oneOverN;
56    }
57
58    public int minimum(int a, int b){
59        int result;
```

```

1 package zue4Wzw;
2
3 public class LittleBee {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int result = 0;
8         result = Demo.minimum(5, 10*2);
9         System.out.println("das ergebnis ist: " + String.valueOf(result));
10        long result22 = Demo.faculty(-3);
11        System.out.println("das ergebnis ist: " + String.valueOf(result22));
12        double result22 = Demo.powerNtWhile(2.0d, 3);
13        System.out.println("das ergebnis ist: " + String.valueOf(result22));
14        double result2 = Demo.exp(2.0d);
15        System.out.println("das ergebnis ist: " + String.valueOf(result2));
16        double[] theArray = new double[3];
17        theArray[0] = 1.0d;
18        theArray[1] = 2.0d;
19        theArray[2] = 3.0d;
20        double result3 = Demo.expectation(theArray);
21        System.out.println("das ergebnis ist: " + String.valueOf(result3));
22        double result4 = Demo.variance(theArray);
23        System.out.println("das ergebnis ist: " + String.valueOf(result4));
24        double[][] matrix = new double[3][3];
25        System.out.println("das ergebnis ist: " + String.valueOf(matrix[2][2]));
26    }
27
28    public static double expectation(double[] a){
29        double result = 0.0d;
30        double oneOverNminusOne = 0.0d;
31        if(a.length >= 1)
32            oneOverNminusOne = 1.0d / ((double)(a.length) - 1);
33        else
34            System.out.println("array is not long enough");
35        double expectation = expectation(a);
36        for(int i=0; i<a.length; i++){
37            result = result + (a[i] - expectation) * (a[i] - expectation);
38        }
39        result = oneOverNminusOne * result;
40        return result;
41    }
42
43    public static int minimum(int a, int b){
44        int result;
45        if(a < b){
46            result = a;
47            //System.out.println("kjxhjhjh");
48        } else
49            result = b;
50        return result;
51    }
52
53    public static long faculty(int a){
54        long result = 1;
55        while(a > 1){
56            result = result * a;
57            a = a - 1;
58        }
59        return result;
60    }
61
62    public static void main() {
63        System.out.println("Hello World!");
64    }
65}

```

```

1 package zue4Wzw;
2
3 public class Demo {
4
5     public static void main() {
6         System.out.println("Hello World!");
7     }
8
9     public static double powerNtWhile(double argument, int m){
10        double result = 1.0d;
11        int i = m;
12        while(i > 0){
13            result = result * argument;
14            i = i - 1;
15        }
16        return result;
17    }
18
19    public static int minimum2(int a, int b){
20        if(a < b)
21            return a;
22        else
23            return b;
24    }
25
26    public static long faculty(int a){
27        long result = 1;
28        while(a > 1){
29            result = result * a;
30            a = a - 1;
31        }
32        return result;
33    }
34
35    public static void main() {
36        System.out.println("Hello World!");
37    }
38}

```

```

1 package zue4Wzw;
2
3 public class Demo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int result = 0;
8         result = Demo.minimum(5, 10*2);
9         System.out.println("das ergebnis ist: " + String.valueOf(result));
10        long result22 = Demo.faculty(-3);
11        System.out.println("das ergebnis ist: " + String.valueOf(result22));
12        double result22 = Demo.powerNtWhile(2.0d, 3);
13        System.out.println("das ergebnis ist: " + String.valueOf(result22));
14        double result2 = Demo.exp(2.0d);
15        System.out.println("das ergebnis ist: " + String.valueOf(result2));
16        double[] theArray = new double[3];
17        theArray[0] = 1.0d;
18        theArray[1] = 2.0d;
19        theArray[2] = 3.0d;
20        double result3 = Demo.expectation(theArray);
21        System.out.println("das ergebnis ist: " + String.valueOf(result3));
22        double result4 = Demo.variance(theArray);
23        System.out.println("das ergebnis ist: " + String.valueOf(result4));
24        double[][] matrix = new double[3][3];
25        System.out.println("das ergebnis ist: " + String.valueOf(matrix[2][2]));
26    }
27
28    public static double expectation(double[] a){
29        double result = 0.0d;
30        double oneOverNminusOne = 0.0d;
31        if(a.length >= 1)
32            oneOverNminusOne = 1.0d / ((double)(a.length) - 1);
33        else
34            System.out.println("array is not long enough");
35        double expectation = expectation(a);
36        for(int i=0; i<a.length; i++){
37            result = result + (a[i] - expectation) * (a[i] - expectation);
38        }
39        result = oneOverNminusOne * result;
40        return result;
41    }
42
43    public static int minimum(int a, int b){
44        int result;
45        if(a < b){
46            result = a;
47            //System.out.println("kjxhjhjh");
48        } else
49            result = b;
50        return result;
51    }
52
53    public static long faculty(int a){
54        long result = 1;
55        while(a > 1){
56            result = result * a;
57            a = a - 1;
58        }
59        return result;
60    }
61
62    public static void main() {
63        System.out.println("Hello World!");
64    }
65}

```

Console Output:

```

terminated> Demo [Data Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 13:56:56)
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0

```

Overloading

Overloading: In einer Klasse mehrere Methoden mit **gleichem Namen**, aber **verschiedener Parameterliste**:

```

class OverloadingDemoClass {
    public int doSomething() {
        return 1 + 1;
    }

    public int doSomething(int param) {
        return param + 2;
    }
}

```

```

OverloadingDemoClass odc = new OverloadingDemoClass();
int result1 = odc.doSomething();
int result2 = odc.doSomething(33);

```

Sinn: Flexibilität (speziellere und weniger spezielle Varianten der Methode anbieten, Abstraktion (→ APIs):

Overloading

Overloading: In einer Klasse mehrere Methoden mit **gleichem Namen**, aber **verschiedener Parameterliste**:

```
class OverloadingDemoClass {  
    public int doSomething() {  
        return 1 + 1;  
    }  
  
    public int doSomething(int param) {  
        return param + 2;  
    }  
}
```

```
OverloadingDemoClass odc = new OverloadingDemoClass();  
int result1 = odc.doSomething();  
int result2 = odc.doSomething(33);
```

Sinn: Flexibilität (speziellere und weniger spezielle Varianten der Methode anbieten, Abstraktion (→ APIs):

Overriding

Overriding: In einer **Unterklasse** Methode mit **gleichem Namen**, und **gleicher Parameterliste** wie in Oberklasse:

```
class Bicycle {  
    int speed;  
    public void speedUp(int increment) {  
        speed = speed + increment;  
        System.out.println("superclass instance-method");  
    }  
  
    class MountainBike extends Bicycle {  
        public void speedUp(int increment) {  
            speed = speed + increment;  
            System.out.println("subclass instance-method");  
        }  
    }  
}  
  
MountainBike mb = new MountainBike();  
mb.speedUp(10); // mb.speed == 20
```

→ Ausgabe: subclass instance-method

Sinn: Unterklasse bietet speziellere Version der Methode an (Aspekt von Polymorphie)

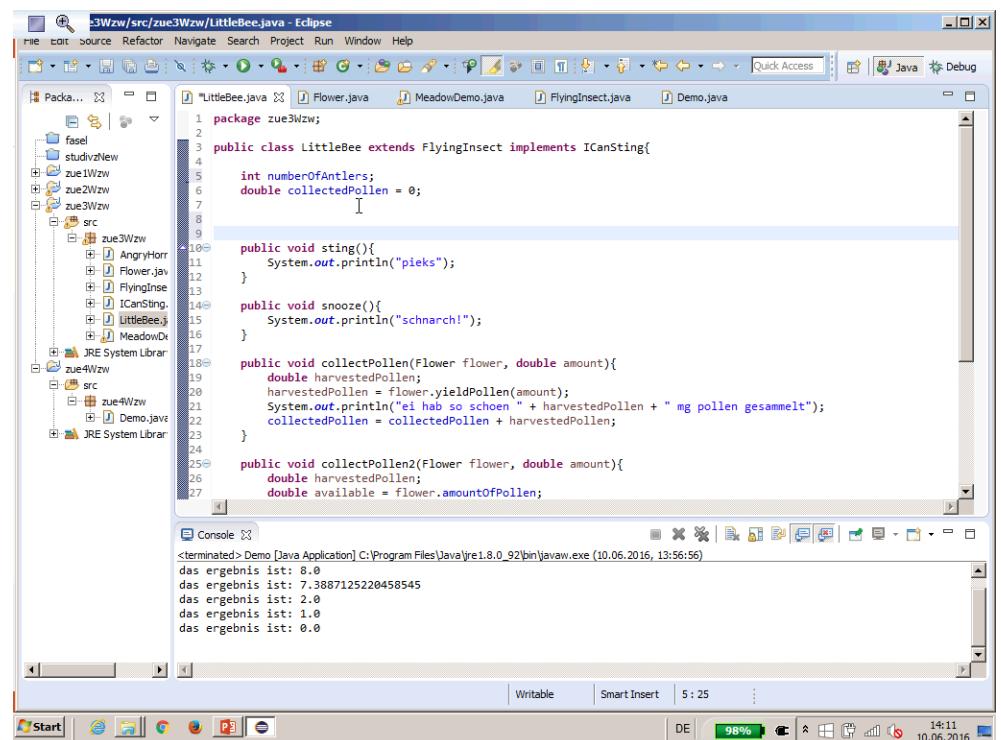
Overriding

Overriding: In einer **Unterklasse** Methode mit **gleichem Namen**, und **gleicher Parameterliste** wie in Oberklasse:

```
class Bicycle {  
    int speed;  
    public void speedUp(int increment) {  
        speed = speed + increment;  
        System.out.println("superclass instance-method");  
    }  
  
    class MountainBike extends Bicycle {  
        public void speedUp(int increment) {  
            super(2 * increment); // call overridden method of superclass  
            System.out.println("subclass instance-method");  
        }  
    }  
}
```

```
MountainBike mb = new MountainBike();  
mb.speedUp(10); // mb.speed == 20
```

→ Ausgabe: superclass instance-method
subclass instance-method



Sinn: Unterklasse bietet speziellere Version der Methode an (Aspekt von Polymorphie)

Eclipse IDE screenshot showing the code for LittleBee.java:

```
package zue3Wzw;
public class LittleBee extends FlyingInsect implements ICanSting{
    int numberofAntlers;
    double collectedPollen = 0;
    public LittleBee(int numberofAntlers, double collectedPollen){
        this.numberofAntlers = numberofAntlers;
        this.collectedPollen = collectedPollen;
    }
    public void sting(){
        System.out.println("pieks");
    }
    public void snooze(){
        System.out.println("schnarch!");
    }
    public void collectPollen(Flower flower, double amount){
        double harvestedPollen;
        harvestedPollen = flower.yieldPollen(amount);
        System.out.println("ei hab so schoen " + harvestedPollen + " mg pollen gesammelt");
        collectedPollen = collectedPollen + harvestedPollen;
    }
    public void collectPollen2(Flower flower, double amount){
    }
}
```

The console output shows the results of running the application:

```
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```

Eclipse IDE screenshot showing the code for LittleBee.java with a minor modification:

```
package zue3Wzw;
public class LittleBee extends FlyingInsect implements ICanSting{
    int numberofAntlers;
    double collectedPollen = 0;
    public LittleBee(int numberofAntlersParam, double collectedPollenParam){
        numberofAntlers = numberofAntlersParam;
        collectedPollen = collectedPollenParam;
    }
    public void sting(){
        System.out.println("pieks");
    }
    public void snooze(){
        System.out.println("schnarch!");
    }
    public void collectPollen(Flower flower, double amount){
        double harvestedPollen;
        harvestedPollen = flower.yieldPollen(amount);
        System.out.println("ei hab so schoen " + harvestedPollen + " mg pollen gesammelt");
        collectedPollen = collectedPollen + harvestedPollen;
    }
    public void collectPollen2(Flower flower, double amount){
    }
}
```

The console output shows the results of running the application:

```
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```

Eclipse IDE screenshot showing the code for LittleBee.java with another modification:

```
package zue3Wzw;
public class LittleBee extends FlyingInsect implements ICanSting{
    int numberofAntlers;
    double collectedPollen = 0;
    public LittleBee(int numberofAntlersParam, double collectedPollenParam){
        numberofAntlers = numberofAntlersParam;
        this.collectedPollen = collectedPollenParam;
    }
    public void sting(){
        System.out.println("pieks");
    }
    public void snooze(){
        System.out.println("schnarch!");
    }
    public void collectPollen(Flower flower, double amount){
        double harvestedPollen;
        harvestedPollen = flower.yieldPollen(amount);
        System.out.println("ei hab so schoen " + harvestedPollen + " mg pollen gesammelt");
        collectedPollen = collectedPollen + harvestedPollen;
    }
    public void collectPollen2(Flower flower, double amount){
    }
}
```

The console output shows the results of running the application:

```
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```

Eclipse IDE screenshot showing the final version of LittleBee.java:

```
package zue3Wzw;
public class LittleBee extends FlyingInsect implements ICanSting{
    int numberofAntlers;
    double collectedPollen = 0;
    public LittleBee(int numberofAntlersParam, double collectedPollenParam){
        numberofAntlers = numberofAntlersParam;
        this.collectedPollen = collectedPollenParam;
    }
    public void sting(){
        System.out.println("pieks");
    }
    public void snooze(){
        System.out.println("schnarch!");
    }
    public void collectPollen(Flower flower, double amount){
        double harvestedPollen;
        harvestedPollen = flower.yieldPollen(amount);
        System.out.println("ei hab so schoen " + harvestedPollen + " mg pollen gesammelt");
        collectedPollen = collectedPollen + harvestedPollen;
    }
    public void collectPollen2(Flower flower, double amount){
    }
}
```

The console output shows the results of running the application:

```
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```

Eclipse IDE screenshot showing the code for LittleBee.java:

```
package zue3Wzw;
public class LittleBee extends FlyingInsect implements ICanSting{
    int numberOfAntlers;
    double collectedPollen = 0;
    public LittleBee(int numberOfAntlersParam, double collectedPollen){
        numberOfAntlers = numberOfAntlersParam;
        this.collectedPollen = collectedPollen;
    }
    public void sting(){
        System.out.println("pieks");
    }
    public void snooze(){
        System.out.println("schnarch!");
    }
    public void collectPollen(Flower flower, double amount){
        double harvestedPollen;
        harvestedPollen = flower.yieldPollen(amount);
        System.out.println("ei hab so schoen " + harvestedPollen + " mg pollen gesammelt");
        collectedPollen = collectedPollen + harvestedPollen;
    }
}
```

The console output shows the following results:

```
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```

Eclipse IDE screenshot showing the code for LittleBee.java:

```
package zue3Wzw;
public class LittleBee extends FlyingInsect implements ICanSting{
    static final int TYPICAL_NO_OF_ANTLERS = 2;
    int numberOfAntlers;
    double collectedPollen = 0;
    public LittleBee(int numberOfAntlersParam, double collectedPollen){
        numberOfAntlers = numberOfAntlersParam;
        this.collectedPollen = collectedPollen;
    }
    public void sting(){
        System.out.println("pieks");
    }
    public void snooze(){
        System.out.println("schnarch!");
    }
    public void collectPollen(Flower flower, double amount){
        double harvestedPollen;
        harvestedPollen = flower.yieldPollen(amount);
        System.out.println("ei hab so schoen " + harvestedPollen + " mg pollen gesammelt");
    }
}
```

The console output shows the following results:

```
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```

Eclipse IDE screenshot showing the code for LittleBee.java:

```
package zue3Wzw;
public class LittleBee extends FlyingInsect implements ICanSting{
    static final int TYPICAL_NO_OF_ANTLERS = 2;
    int numberOfAntlers;
    double collectedPollen = 0;
    public LittleBee(int numberOfAntlersParam, double collectedPollen){
        numberOfAntlers = numberOfAntlersParam;
        this.collectedPollen = collectedPollen;
    }
    public LittleBee(double collectedPollen){
        numberOfAntlers = TYPICAL_NO_OF_ANTLERS;
        this.collectedPollen = collectedPollen;
    }
    public void sting(){
        System.out.println("pieks");
    }
    public void snooze(){
        System.out.println("schnarch!");
    }
}
```

The console output shows the following results:

```
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```

Eclipse IDE screenshot showing the code for LittleBee.java:

```
package zue3Wzw;
public class LittleBee extends FlyingInsect implements ICanSting{
    static final int TYPICAL_NO_OF_ANTLERS = 2;
    int numberOfAntlers;
    double collectedPollen = 0;
    public LittleBee(int numberOfAntlersParam, double collectedPollen){
        numberOfAntlers = numberOfAntlersParam;
        this.collectedPollen = collectedPollen;
    }
    public LittleBee(double collectedPollen){
        numberOfAntlers = TYPICAL_NO_OF_ANTLERS;
        this.collectedPollen = collectedPollen;
    }
    public void sting(){
        System.out.println("pieks");
    }
    public void snooze(){
        System.out.println("schnarch!");
    }
}
```

The console output shows the following results:

```
das ergebnis ist: 8.0
das ergebnis ist: 7.3887125220458545
das ergebnis ist: 2.0
das ergebnis ist: 1.0
das ergebnis ist: 0.0
```

-3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

File Edit Source Refactor Navigate Project Run Window Help

LittleBee.java Flower.java *MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(1, 20.0d);
9         LittleBee willi = new LittleBee();
10        maja.fly();
11        maja.sting();
12        willi.snooze();
13        AngryHornet evil;
14        evil = new AngryHornet();
15        evil.sting();
16
17        ICanSting someStinger;
18        someStinger = evil;
19        someStinger.sting();
20
21        Flower flower1 = new Flower();
22        Flower flower2 = new Flower();
23
24        maja.collectPollen(flower1, 20.0d);
25
26    }
27
28 }
```

-3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

File Edit Source Refactor Navigate Project Run Window Help

LittleBee.java Flower.java MeadowDemo.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee();
10        maja.fly();
11        maja.sting();
12        willi.snooze();
13        AngryHornet evil;
14        evil = new AngryHornet();
15        evil.sting();
16
17        ICanSting someStinger;
18        someStinger = evil;
19        someStinger.sting();
20
21        Flower flower1 = new Flower();
22        Flower flower2 = new Flower();
23
24        maja.collectPollen(flower1, 20.0d);
25
26    }
27
28 }
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:17:28)
brunna
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/LittleBee.java - Eclipse

File Edit Source Refactor Navigate Project Run Window Help

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class LittleBee extends FlyingInsect implements ICanSting{
4
5     static final int TYPICAL_NO_OF_ANTLERS = 2;
6     int numberOfAntlers;
7     double collectedPollen = 0;
8
9     public LittleBee(int numberOfAntlersParam, double collectedPollen){
10        numberOfAntlers = numberOfAntlersParam;
11        this.collectedPollen = collectedPollen;
12    }
13
14    public LittleBee(double collectedPollen){
15        numberOfAntlers = TYPICAL_NO_OF_ANTLERS;
16        this.collectedPollen = collectedPollen;
17    }
18
19    public void fly(){ String arg0}
20    System.out.println(arg0);
21    arg0
22
23    public void sting(){
24        System.out.println("pieks");
25    }
26
27    public void snooze(){
28        System.out.println("schnarch!");
29    }
30
31    public void collectPollen(Flower flower, double amount){
32        double harvestedPollen;
33        harvestedPollen = flower.yieldPollen(amount);
34        System.out.println("ei hab so schoen " + harvestedPollen + " mg pollen gesammelt");
35        collectedPollen = collectedPollen + harvestedPollen;
36    }
37
38    public void collectPollen2(Flower flower, double amount){
39        ...
40    }
41 }
```

Java - zue3Wzw/src/zue3Wzw/LittleBee.java - Eclipse

File Edit Source Refactor Navigate Project Run Window Help

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class LittleBee extends FlyingInsect implements ICanSting{
4
5     static final int TYPICAL_NO_OF_ANTLERS = 2;
6     int numberOfAntlers;
7     double collectedPollen = 0;
8
9     public LittleBee(int numberOfAntlersParam, double collectedPollen){
10        numberOfAntlers = numberOfAntlersParam;
11        this.collectedPollen = collectedPollen;
12    }
13
14    public LittleBee(double collectedPollen){
15        numberOfAntlers = TYPICAL_NO_OF_ANTLERS;
16        this.collectedPollen = collectedPollen;
17    }
18
19    public void fly(){ I
20        System.out.println("bienenbrumm");
21    }
22
23    public void sting(){
24        System.out.println("pieks");
25    }
26    ...
27 }
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:18:57)
bienenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/LittleBee.java - Eclipse

```
File Edit Source Refactor Navigate Search Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class LittleBee extends FlyingInsect implements ICanSing{
4
5     static final int TYPICAL_NO_OF_ANTLERS = 2;
6     int numberOfAntlers;
7     double collectedPollen = 0;
8
9     public LittleBee(int numberOfAntlersParam, double collectedPollen){
10        numberOfAntlers = numberOfAntlersParam;
11        this.collectedPollen = collectedPollen;
12    }
13
14    public LittleBee(double collectedPollen){
15        numberOfAntlers = TYPICAL_NO_OF_ANTLERS;
16        this.collectedPollen = collectedPollen;
17    }
18
19    public void fly(){
20        super();
21        System.out.println("bienenbrumm");
22    }
23
24    public void sting(){
25        System.out.println("pieks");
26    }
27}
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:19:34)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Constructor call must be the first statement in a constructor

at zue3Wzw.LittleBee.fly(LittleBee.java:20)
at zue3Wzw.MeadowDemo.main(MeadowDemo.java:10)
```

Start

Java - zue3Wzw/src/zue3Wzw/LittleBee.java - Eclipse

```
File Edit Source Refactor Navigate Search Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class LittleBee extends FlyingInsect implements ICanSing{
4
5     static final int TYPICAL_NO_OF_ANTLERS = 2;
6     int numberOfAntlers;
7     double collectedPollen = 0;
8
9     public LittleBee(int numberOfAntlersParam, double collectedPollen){
10        numberOfAntlers = numberOfAntlersParam;
11        this.collectedPollen = collectedPollen;
12    }
13
14    public LittleBee(double collectedPollen){
15        numberOfAntlers = TYPICAL_NO_OF_ANTLERS;
16        this.collectedPollen = collectedPollen;
17    }
18
19    public void fly(){
20        this.super();
21        System.out.println("bienenbrumm");
22    }
23
24    public void sting(){
25        System.out.println("pieks");
26    }
27}
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:19:59)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Constructor call must be the first statement in a constructor

at zue3Wzw.LittleBee.fly(LittleBee.java:20)
at zue3Wzw.MeadowDemo.main(MeadowDemo.java:10)
```

Start

Java - zue3Wzw/src/zue3Wzw/LittleBee.java - Eclipse

```
File Edit Source Refactor Navigate Search Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class LittleBee extends FlyingInsect implements ICanSing{
4
5     static final int TYPICAL_NO_OF_ANTLERS = 2;
6     int numberOfAntlers;
7     double collectedPollen = 0;
8
9     public LittleBee(int numberOfAntlersParam, double collectedPollen){
10        numberOfAntlers = numberOfAntlersParam;
11        this.collectedPollen = collectedPollen;
12    }
13
14    public LittleBee(double collectedPollen){
15        numberOfAntlers = TYPICAL_NO_OF_ANTLERS;
16        this.collectedPollen = collectedPollen;
17    }
18
19    public void fly(){
20        System.out.println("bienenbrumm");
21    }
22
23    public void sting(){
24        System.out.println("pieks");
25    }
26}
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:20:30)
bienenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Start

Java - zue3Wzw/src/zue3Wzw/LittleBee.java - Eclipse

```
File Edit Source Refactor Navigate Search Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class LittleBee extends FlyingInsect implements ICanSing{
4
5     static final int TYPICAL_NO_OF_ANTLERS = 2;
6     int numberOfAntlers;
7     double collectedPollen = 0;
8
9     public LittleBee(int numberOfAntlersParam, double collectedPollen){
10        numberOfAntlers = numberOfAntlersParam;
11        this.collectedPollen = collectedPollen;
12    }
13
14    public LittleBee(double collectedPollen){
15        numberOfAntlers = TYPICAL_NO_OF_ANTLERS;
16        this.collectedPollen = collectedPollen;
17    }
18
19    public void fly(){
20        System.out.println("bienenbrumm");
21    }
22
23    public void sting(){
24        System.out.println("pieks");
25    }
26}
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:20:30)
bienenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Start

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d); [
10
11         [
12             maja.fly();
13             maja.sting();
14             willi.snooze();
15             AngryHornet evil;
16             evil = new AngryHornet();
17             evil.sting();
18
19             ICanSting someStinger;
20             someStinger = evil;
21             someStinger.sting();
22
23             Flower flower1 = new Flower();
24             Flower flower2 = new Flower();
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:20:30)
bielenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java *MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10         FlyingInsect heiner = new FlyingInsect; [
11
12             maja.fly();
13             maja.sting();
14             willi.snooze();
15             AngryHornet evil;
16             evil = new AngryHornet();
17             evil.sting();
18
19             ICanSting someStinger;
20             someStinger = evil;
21             someStinger.sting();
22
23             Flower flower1 = new Flower();
24             Flower flower2 = new Flower();
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:20:30)
bielenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10         FlyingInsect heiner = new FlyingInsect();
11         heiner.fly(); [
12
13             maja.fly();
14             maja.sting();
15             willi.snooze();
16             AngryHornet evil;
17             evil = new AngryHornet();
18             evil.sting();
19
20             ICanSting someStinger;
21             someStinger = evil;
22             someStinger.sting();
23
24             Flower flower1 = new Flower();
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:24:26)
standardbrumm
bielenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java *MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10         FlyingInsect heiner = new FlyingInsect();
11         heiner.fly(); [
12             heiner = maja; [
13                 maja.fly(); [
14                     maja.sting();
15                     willi.snooze();
16                     AngryHornet evil;
17                     evil = new AngryHornet();
18                     evil.sting();
19
20                     ICanSting someStinger;
21                     someStinger = evil;
22                     someStinger.sting();
23
24                     Flower flower1 = new Flower();
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:24:26)
standardbrumm
bielenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java *MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10        FlyingInsect heiner = new FlyingInsect();
11        heiner.fly();
12        heiner = maja;
13        maja.fly();
14        maja.sting();
15        willi.snooze();
16        AngryHornet evil = new AngryHornet();
17        evil = maja;
18        evil.sting();
19
20        ICanSting someStinger;
21        someStinger = evil;
22        someStinger.sting();
23
24        Flower flower1 = new Flower();
}
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:24:26)
standardbrumm
bienenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10        FlyingInsect heiner = new FlyingInsect();
11        heiner.fly();
12        heiner = maja;
13        heiner.fly();
14        //maja.fly();
15        maja.sting();
16        willi.snooze();
17        AngryHornet evil;
18        evil = new AngryHornet();
19        evil.sting();
20
21        ICanSting someStinger;
22        someStinger = evil;
23        someStinger.sting();
24}
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:25:58)
standardbrumm
bienenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10        FlyingInsect heiner = new FlyingInsect();
11        heiner.fly();
12        heiner = maja;
13        heiner.fly(); I
14        //maja.fly();
15        maja.sting();
16        willi.snooze();
17        AngryHornet evil;
18        evil = new AngryHornet();
19        evil.sting();
20
21        ICanSting someStinger;
22        someStinger = evil;
23        someStinger.sting();
24}
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:25:58)
standardbrumm
bienenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java *MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d); ..
10        FlyingInsect heiner = new FlyingInsect();
11        FlyingInsect reserveReferenceToHeiner = heiner;
12        heiner.fly();
13        heiner = maja;
14        heiner.fly();
15        //maja.fly();
16        maja.sting();
17        willi.snooze();
18        AngryHornet evil;
19        evil = new AngryHornet();
20        evil.sting();
21
22        ICanSting someStinger;
23        someStinger = evil;
24        someStinger.sting();
25}
```

Console

```
<terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:25:58)
standardbrumm
bienenbrumm
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10        FlyingInsect heiner = new FlyingInsect();
11        FlyingInsect reserveReferenceToHeiner = heiner;
12        heiner.fly();
13        heiner.fly();
14        heiner.fly();
15        heiner.fly();
16        heiner.fly();
17        //maja.fly();
18        maja.sting();
19        willi.snooze();
20        AngryHornet evil;
21        ...
22    }
23}
```

Console <terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:30:03)

```
standardbrunn
bienenbrunn
standardbrunn
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10        FlyingInsect heiner = new FlyingInsect();
11        FlyingInsect reserveReferenceToHeiner = heiner;
12        heiner.fly();
13        heiner.fly();
14        heiner.fly();
15        heiner.fly();
16        heiner.fly();
17        //maja.fly();
18        maja.sting();
19        willi.snooze();
20        AngryHornet evil;
21        ...
22    }
23}
```

Console <terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:30:03)

```
standardbrunn
bienenbrunn
standardbrunn
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10        FlyingInsect heiner = new FlyingInsect();
11        FlyingInsect reserveReferenceToHeiner = heiner;
12        heiner.fly();
13        heiner.fly();
14        heiner.fly();
15        heiner.fly();
16        heiner.fly();
17        maja.fly(20.0d);
18        maja.sting();
19        willi.snooze();
20        AngryHornet evil;
21        ...
22    }
23}
```

Console <terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:30:03)

```
standardbrunn
bienenbrunn
standardbrunn
bienenbrunn, und zwar sooo schnell: 20.0
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue3Wzw/src/zue3Wzw/MeadowDemo.java - Eclipse

```
File Edit Source Refactor Navigate Project Run Window Help
```

LittleBee.java Flower.java MeadowDemo.java FlyingInsect.java

```
1 package zue3Wzw;
2
3 public class MeadowDemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         LittleBee maja;
8         maja = new LittleBee(2, 20.0d);
9         LittleBee willi = new LittleBee(3, 30.0d);
10        FlyingInsect heiner = new FlyingInsect();
11        FlyingInsect reserveReferenceToHeiner = heiner;
12        heiner.fly();
13        heiner.fly();
14        heiner.fly();
15        heiner.fly();
16        heiner.fly();
17        maja.fly(20.0d);
18        maja.sting();
19        willi.snooze();
20        AngryHornet evil;
21        evil = new AngryHornet();
22        evil.sting();
23
24        ICanSting someStinger;
25        someStinger = evil;
26        someStinger.sting();
27
28        Flower flower1 = new Flower();
29        Flower flower2 = new Flower();
30
31        maja.collectPollen(flower1, 20.0d);
32
33    }
34
35}
36
```

Console <terminated> MeadowDemo [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:31:22)

```
standardbrunn
bienenbrunn
standardbrunn
bienenbrunn, und zwar sooo schnell: 20.0
pieks
schnarch!
MEGA-pieks
MEGA-pieks
uebrige pollen: 80.0
ei hab so schoen 20.0 mg pollen gesammelt
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6     }
7
8 }
9
10 }
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6     }
7
8 }
9
10 }
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4     public static void bubbleSort() {
5
6
7     }
8
9     public static void main(String[] args) {
10         // TODO Auto-generated method stub
11     }
12
13 }
14
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4     public static void bubbleSort(int[] data) {
5
6
7     }
8
9     public static void main(String[] args) {
10         // TODO Auto-generated method stub
11     }
12
13 }
14
15 }
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(int[] data){
6         for(int i = data.length-1; i>0; i--){
7             for(int j=0; j<i; j++){
8                 if(data[j] > data[j+1]){
9                     //swap elements
10                    data[j] = data[j+1];
11                    data[j+1] = data[j];
12                }
13            }
14        }
15    }
16
17    public static void main(String[] args) {
18        // TODO Auto-generated method stub
19    }
20}
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(int[] data){
6         for(int i = data.length-1; i>0; i--){
7             for(int j=0; j<i; j++){
8                 if(data[j] > data[j+1]){
9                     //swap elements
10                    data[j] = data[j+1];
11                    data[j+1] = data[j];
12                }
13            }
14        }
15    }
16
17    public static void main(String[] args) {
18        // TODO Auto-generated method stub
19    }
20}
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(int[] data){
6         for(int i = data.length-1; i>0; i--){
7             for(int j=0; j<i; j++){
8                 if(data[j] > data[j+1]){
9                     //swap elements
10                    data[j] = data[j+1];
11                    data[j+1] = data[j];
12                }
13            }
14        }
15    }
16
17    public static void main(String[] args) {
18        // TODO Auto-generated method stub
19    }
20}
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(int[] data){
6         for(int i = data.length-1; i>0; i--){
7             for(int j=0; j<i; j++){
8                 if(data[j] > data[j+1]){
9                     //swap elements
10                    backup = data[j];
11                    data[j] = data[j+1];
12                    data[j+1] = backup;
13                }
14            }
15        }
16    }
17
18    public static void main(String[] args) {
19        // TODO Auto-generated method stub
20        int[] foo = {5,3,7,1,2,10};
21        showArray(foo);
22    }
23
24    public static void showArray(int[] theArray){
25        for(int i=0; i<theArray.length; i++){
26            System.out.print(" " + theArray[i]);
27            System.out.println();
28        }
29    }
30
31    public static void main(String[] args) {
32        // TODO Auto-generated method stub
33    }
34}
```

Console

```
>InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:46:29)
, 5
, 3
, 7
, 1
, 2
, 10
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
public static void bubbleSort(int[] data){  
    int backup;  
    for(int i = data.length-1; i>0; i--){  
        for(int j=0; j<i; j++){  
            if(data[j]> data[j+1]){  
                //swap elements  
                backup = data[j];  
                data[j] = data[j+1];  
                data[j+1] = backup;  
            }  
        }  
    }  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int[] foo = {5,3,7,1,2,10};  
        showArray(foo);  
    }  
  
    public static void showArray(int[] theArray){  
        for(int i=0; i<theArray.length; i++){  
            System.out.print(theArray[i] + " ");  
        }  
    }  
}
```

Console

```
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:47:07)  
5, 3, 7, 1, 2, 10,
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
public static void bubbleSort(int[] data){  
    int backup;  
    for(int i = data.length-1; i>0; i--){  
        for(int j=0; j<i; j++){  
            if(data[j]> data[j+1]){  
                //swap elements  
                backup = data[j];  
                data[j] = data[j+1];  
                data[j+1] = backup;  
            }  
        }  
    }  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int[] foo = {5,3,7,1,2,10};  
        showArray(foo);  
        bubbleSort(foo);  
        showArray(foo);  
    }  
  
    public static void showArray(int[] theArray){  
        for(int i=0; i<theArray.length; i++){  
            System.out.print(theArray[i] + " ");  
        }  
    }  
}
```

Console

```
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:48:20)  
5, 3, 7, 1, 2, 10,  
1, 2, 3, 5, 7, 10,
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

New Java Interface

Java Interface

Create a new Java interface.

Main.java

```
package zue5Wzw;  
public interface ICanBeCompared {  
}
```

Source folder: zue5Wzw/src

Package: zue5Wzw

Encl type: zue5Wzw.InterfaceDemoMain

Name: ICanBeCompared

Modifiers: public

Extended interfaces:

Do you want to add comments? (Configure templates and default value [here](#))
 Generate comments

Finish Cancel

Console

```
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:48:20)  
5, 3, 7, 1, 2, 10,  
1, 2, 3, 5, 7, 10,
```

Java - zue5Wzw/src/zue5Wzw/ICanBeCompared.java - Eclipse

```
package zue5Wzw;  
public interface ICanBeCompared {  
}
```

Console

```
<terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:48:20)  
5, 3, 7, 1, 2, 10,  
1, 2, 3, 5, 7, 10,
```

Java - zue5Wzw/src/zue5Wzw/ICanBeCompared.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

New Open File... Save Save As... Save All Rename... Refresh Convert Line Delimiters To Print... Switch Workspace Restart Import... Export... Properties Alt+Enter

1 ICanBeCompared.java [zue5Wzw/src/...] 2 InterfaceDemoMain.java [zue5Wzw/] 3 LittleBee.java [zue3Wzw/src/zue3Wzw] 4 Demo.java [zue4Wzw/src/zue4Wzw]

Exit

ICanBeCompared {
 isGreaterThan();}

Console <terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:48:20)
5, 3, 7, 1, 2, 10,
1, 2, 3, 5, 7, 10,

Start | Writable | Smart Insert | 5 : 36 | DE | 98% | 14:50 | 10.06.2016

Java - zue5Wzw/src/zue5Wzw/Student.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access Java Debug

Packag... LittleBee.java MeadowDemo.java FlyingInsect... InterfaceDe... ICanBeCompared... Student.java

1 package zue5Wzw;
2
3 public class Student implements ICanBeCompared {
4
5 public String firstName;
6 public int matriculationNo;
7
8 public Student(String someName, int someNo){
9 firstName = someName;
10 matriculationNo = someNo;
11 }
12
13 public boolean isGreaterThan(ICanBeCompared theOtherThing){
14 if(theOtherThing instanceof Student){
15 return true;
16 }
17 }
18
19
20 }

Console <terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:48:20)
5, 3, 7, 1, 2, 10,
1, 2, 3, 5, 7, 10,

Syntax error on token ")", Statement expected after this token

Start | Writable | Smart Insert | 14 : 26 | DE | 98% | 14:56 | 10.06.2016

Java - zue5Wzw/src/zue5Wzw/Student.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access Java Debug

Packag... LittleBee.java MeadowDemo.java FlyingInsect... InterfaceDe... ICanBeCompared... Student.java

1 package zue5Wzw;
2
3 public class Student implements ICanBeCompared {
4
5 public String firstName;
6 public int matriculationNo;
7
8 public Student(String someName, int someNo){
9 firstName = someName;
10 matriculationNo = someNo;
11 }
12
13 public boolean isGreaterThan(ICanBeCompared theOtherThing){
14 if(theOtherThing instanceof Student){
15 return true;
16 }
17 }
18
19
20 }

Console <terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:48:20)
5, 3, 7, 1, 2, 10,
1, 2, 3, 5, 7, 10,

Start | Writable | Smart Insert | 14 : 26 | DE | 98% | 14:56 | 10.06.2016

Java - zue5Wzw/src/zue5Wzw/Student.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access Java Debug

Packag... LittleBee.java MeadowDemo.java FlyingInsect... InterfaceDe... ICanBeCompared... Student.java

1 package zue5Wzw;
2
3 public class Student implements ICanBeCompared {
4
5 public String firstName;
6 public int matriculationNo;
7
8 public Student(String someName, int someNo){
9 firstName = someName;
10 matriculationNo = someNo;
11 }
12
13 public boolean isGreaterThan(ICanBeCompared theOtherThing){
14 boolean result;
15 if(theOtherThing instanceof Student){
16 result = ((Student)theOtherThing).matriculationNo
17 }
18 return result;
19 }
20
21
22 }

Console <terminated> InterfaceDemoMain [Java Application] C:\Program Files\Java\jre1.8.0_92\bin\javaw.exe (10.06.2016, 14:48:20)
5, 3, 7, 1, 2, 10,
1, 2, 3, 5, 7, 10,

Start | Writable | Smart Insert | 16 : 62 | DE | 98% | 14:57 | 10.06.2016

Java - zue5Wzw/src/zue5Wzw/Student.java - Eclipse

```
1 package zue5Wzw;
2
3 public class Student implements ICanBeCompared {
4
5     public String firstName;
6     public int matriculationNo;
7
8     public Student(String someName, int someNo){
9         firstName = someName;
10        matriculationNo = someNo;
11    }
12
13    public boolean isGreaterThan(ICanBeCompared theOtherThing){
14        boolean result = false;
15        if(theOtherThing instanceof Student){
16            result = ((Student)theOtherThing).matriculationNo < this.matriculationNo;
17        } else {
18            System.err.println("Comparison between Student and " + theOtherThing.getClass().getName() + " is not supported");
19        }
20        return result;
21    }
22
23 }
24
25 }
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(int[] data){
6         ICanBeCompared backup;
7         for(int i = data.length-1; i>0; i--){
8             for(int j=0; j<i; j++){
9                 if(data[j].isGreaterThan(data[j+1])){
10                     //swap elements
11                     backup = data[j];
12                     data[j] = data[j+1];
13                     data[j+1] = backup;
14                 }
15             }
16         }
17     }
18
19     public static void main(String[] args) {
20         // TODO Auto-generated method stub
21         int[] foo = {5,3,7,1,2,10};
22         showArray(foo);
23         bubbleSort(foo);
24         Student[] someStudents = new Student[2];
25         someStudents[0] = new Student("heiner", 5000);
26         someStudents[1] = new Student("horst", 4000);
27         someStudents[0].isGreaterThan(someStudents[1]);
28     }
29 }
```

Console

```
5, 3, 7, 1, 2, 10,
1, 2, 3, 5, 7, 10,
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(int[] data){
6         ICanBeCompared backup;
7         for(int i = data.length-1; i>0; i--){
8             for(int j=0; j<i; j++){
9                 if(data[j]> data[j+1]){
10                     //swap elements
11                     backup = data[j];
12                     data[j] = data[j+1];
13                     data[j+1] = backup;
14                 }
15             }
16         }
17     }
18
19     public static void main(String[] args) {
20         // TODO Auto-generated method stub
21         int[] foo = {5,3,7,1,2,10};
22         showArray(foo);
23         bubbleSort(foo);
24         showArray(foo);
25     }
26 }
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(int[] data){
6         ICanBeCompared backup;
7         for(int i = data.length-1; i>0; i--){
8             for(int j=0; j<i; j++){
9                 if(data[j]> data[j+1]){
10                     //swap elements
11                     backup = data[j];
12                     data[j] = data[j+1];
13                     data[j+1] = backup;
14                 }
15             }
16         }
17     }
18
19     public static void main(String[] args) {
20         // TODO Auto-generated method stub
21         int[] foo = {5,3,7,1,2,10};
22         showArray(foo);
23         bubbleSort(foo);
24         showArray(foo);
25     }
26 }
```

Console

```
5, 3, 7, 1, 2, 10,
1, 2, 3, 5, 7, 10,
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(ICanBeCompared[] data){
6         ICanBeCompared backup;
7         for(int i = data.length-1; i>0; i--){
8             for(int j=0; j<i; j++){
9                 if(data[j].isGreaterThan(data[j+1])){
10                     //swap elements
11                     backup = data[j];
12                     data[j] = data[j+1];
13                     data[j+1] = backup;
14                 }
15             }
16         }
17     }
18
19     public static void main(String[] args) {
20         // TODO Auto-generated method stub
21         int[] foo = {5,3,7,1,2,10};
22         showArray(foo);
23         bubbleSort(foo);
24         Student[] someStudents = new Student[2];
25         someStudents[0] = new Student("heiner", 5000);
26         someStudents[1] = new Student("horst", 4000);
27         someStudents[0].isGreaterThan(someStudents[1]);
28     }
29 }
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(ICanBeCompared[] data){
6         ICanBeCompared backup;
7         for(int i = data.length-1; i>0; i--){
8             for(int j=0; j<i; j++){
9                 if(data[j].isGreaterThan(data[j+1])){
10                     //swap elements
11                     backup = data[j];
12                     data[j] = data[j+1];
13                     data[j+1] = backup;
14                 }
15             }
16         }
17     }
18
19     public static void main(String[] args) {
20         // TODO Auto-generated method stub
21         int[] foo = {5,3,7,1,2,10};
22         showArray(foo);
23         bubbleSort(foo);
24         Student[] someStudents = new Student[2];
25         someStudents[0] = new Student("heiner", 5000);
26         someStudents[1] = new Student("horst", 4000);
27         someStudents[0].isGreaterThan(someStudents[1]);
28     }
29 }
```

Console

```
5, 3, 7, 1, 2, 10,
1, 2, 3, 5, 7, 10,
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(ICanBeCompared[] data){
6         ICanBeCompared backup;
7         for(int i = data.length-1; i>0; i--){
8             for(int j=0; j<i; j++){
9                 if(data[j].isGreaterThan(data[j+1])){
10                     //swap elements
11                     backup = data[j];
12                     data[j] = data[j+1];
13                     data[j+1] = backup;
14                 }
15             }
16         }
17     }
18
19     public static void main(String[] args) {
20         // TODO Auto-generated method stub
21         int[] foo = {5,3,7,1,2,10};
22         showArray(foo);
23         bubbleSort(foo);
24         Student[] someStudents = new Student[2];
25         someStudents[0] = new Student("heiner", 5000);
26         someStudents[1] = new Student("horst", 4000);
27         someStudents[0].isGreaterThan(someStudents[1]);
28     }
29 }
```

Java - zue5Wzw/src/zue5Wzw/InterfaceDemoMain.java - Eclipse

```
1 package zue5Wzw;
2
3 public class InterfaceDemoMain {
4
5     public static void bubbleSort(ICanBeCompared[] data){
6         ICanBeCompared backup;
7         for(int i = data.length-1; i>0; i--){
8             for(int j=0; j<i; j++){
9                 if(data[j].isGreaterThan(data[j+1])){
10                     //swap elements
11                     backup = data[j];
12                     data[j] = data[j+1];
13                     data[j+1] = backup;
14                 }
15             }
16         }
17     }
18
19     public static void main(String[] args) {
20         // TODO Auto-generated method stub
21         int[] foo = {5,3,7,1,2,10};
22         showArray(foo);
23         bubbleSort(foo);
24         Student[] someStudents = new Student[2];
25         someStudents[0] = new Student("heiner", 5000);
26         someStudents[1] = new Student("horst", 4000);
27         someStudents[0].isGreaterThan(someStudents[1]);
28     }
29 }
```

Console

```
5, 3, 7, 1, 2, 10,
1, 2, 3, 5, 7, 10,
```

