

Static or What?



```
class TestClass {
    static int calls = 0;
    int count;
    public TestClass(int start){
        count = start;
    }
    int Next() {
        calls++;
        return count++;
    }
    static int Calls() {
        return calls;
    }
}
```

```
TestClass c1 = new TestClass(5);
TestClass c2 = new TestClass(10);
while (c1.Next() < 10);
while (c2.Next() < 20);
System.out.println(
    c1.Next() + " " + c2.Next());
System.out.println(
    c1.Calls() + " " + c2.Calls());
```

Ausgabe ?

- | | |
|-----------------|-----------------|
| (1) 11 21 19 19 | (5) 11 21 17 17 |
| (2) 10 20 5 10 | (6) 10 20 6 11 |
| (3) 11 21 5 10 | (7) 11 21 6 11 |
| (4) 10 20 20 20 | (8) 10 20 17 17 |

Erklärung



```
class TestClass {  
    static int calls = 0;  
    int count;  
    public TestClass(int start){  
        count = start;  
    }  
    int Next() {  
        calls++;  
        return count++;  
    }  
    static int Calls() {  
        return calls;  
    }  
}
```

Statische Variable (auch "Klassenvariable"): nicht an die jeweilige Objektinstanz gebunden. Insofern "global".

Nicht-statische Variable: Zustand des jeweiligen Objektes / der jeweiligen Instanz.

Statische Methode: Methode, welche höchstens of statische Variablen zugreift.

Statische Methode kann mit Instanz- oder Klassennamen angesprochen werden:

```
TestClass c;  
c.Calls() ↔ TestClass.Calls()
```

Erklärung



```
TestClass c1 = new TestClass(5); // c1 beginnt mit Zustand c1.count = 5
TestClass c2 = new TestClass(10); // c2 beginnt mit Zustand c2.count = 10

while (c1.Next() < 10); // c1.count++ bis c1.count++ == 10 (→ c1.i == 11)

while (c2.Next() < 20); // c2.count++ bis c2.count++ == 11 (→ c2.i == 21)

System.out.println(
    c1.Next() + " " + c2.Next()); // 11, 21

System.out.println(
    c1.Calls() + " " + c2.Calls()); // Globale Variable calls wurde bei jedem Aufruf von Next()
// erhöht. Next() gibt immer den vorigen Wert zurück
// → 6 + 11 + 1 + 1 = 19, 19
```

Static or What?



```
class TestClass {
    static int calls = 0;
    int count;
    public TestClass(int start){
        count = start;
    }
    int Next() {
        calls++;
        return count++;
    }
    static int Calls() {
        return calls;
    }
}
```

```
TestClass c1 = new TestClass(5);
TestClass c2 = new TestClass(10);
while (c1.Next() < 10);
while (c2.Next() < 20);
System.out.println(
    c1.Next() + " " + c2.Next());
System.out.println(
    c1.Calls() + " " + c2.Calls());
```

Ausgabe ?

(1) 11 21 19 19	←	(5) 11 21 17 17
(2) 10 20 5 10		(6) 10 20 6 11
(3) 11 21 5 10		(7) 11 21 6 11
(4) 10 20 20 20		(8) 10 20 17 17