
FINAL

Flexibilität vermindern mit final

- Vererbung ermöglicht Erweiterung, aber das könnte teilweise auch unerwünscht sein
- Verhindern von Änderung oder Erweiterung unter anderem durch **final**
- Wirkung möglich auf
 - Klasse
 - Methoden
 - Variablen
 - Attribute

KLASSE ALS FINAL

final bei Klassen

Verhindern der Vererbung

- Klasse mit final darf keine Unterklassen haben
- wird zum Zeitpunkt der Kompilierung erkannt

Verwendungsbeispiele final

- String, Integer, Double, etc. (mgl. Nachteile durch „unkontrollierte“ Vererbung)
- Math (Bibliothek für math. Funktionen)

Hinweis

- genaue Überlegung bei final-Klassen
- andere Programmierer können nie erweitern!
- Objektorientierung wird eingeschränkt

Änderungen im Objekt der Klasse sind aber möglich

Vererbung wird verboten

```
public final class MyFinalClass {  
    private int attribute;  
    // ...  
}
```

```
public class FinalClassSub extends MyFinalClass {  
    // ...  
}
```

Führt zu Compilerfehler:

“The type FinalClassSub cannot subclass the final class MyFinalClass”

```
MyFinalClass eineFinalKlasse = new MyFinalClass(5);  
eineFinalKlasse.setAttribute(8);
```

METHODE ALS FINAL

final bei Methoden /1

Verhindern des Überschreibens

- Methode mit final darf **nicht überschrieben** werden
- wird zum Zeitpunkt der Kompilierung erkannt

Verwendungsbeispiele final

- Thread kann Unterklasse haben, aber die Methode isAlive() darf nicht überschrieben werden (native Methode mit plattformabhängigem Programmcode)

Hinweis

- falls Methode intern in Oberklasse aufgerufen wird, dann führt Überschreiben zum Verändern des Verhalten, **ggf. falsches Verhalten**
- Methoden, die durch Konstruktor aufgerufen werden, sollten final sein (siehe vorheriger Punkt)

Überschreiben wird verboten

```
public class MyFooClass {  
    public final int calculateSquare(int number) {  
        return number*number;  
    }  
}
```

```
public class FooClassSub extends MyFooClass {  
    public int calculateSquare(int number) {  
        return number*number;  
    }  
}
```

Führt zu Compilerfehler:
“Cannot override the final method from MyFooClass”

final bei Methoden /2

final-Klasse vs. final-Methode

- final-Klasse
 - Unterklasse nicht möglich

- final-Methoden, keine final-Klasse
 - Unterklasse möglich
 - Erweiterung möglich
 - final-Methoden bleiben unverändert

```
public final class MyFinalClass {  
    private int attribute;  
    // ...  
}
```

```
public class MyFooClass {  
    public final int calculateSquare(int number) {  
        return number*number;  
    }  
}
```

VARIABLE ALS FINAL

final bei primitiven Variablen

primitive Variable final

- primitiver Datentyp
- Zuweisung einmalig
- Änderung mit weiterer Zuweisung nicht möglich
- Compiler erkennt den Fehler und bricht ab

```
public void finaleVariableNurEinMalZuweisbar() {  
    final int eineVariable;  
    eineVariable = 3;  
    eineVariable = 4;  
}
```

Führt zu Compilerfehler:
**“The final local variable eineVariable
may already have been assigned”**

final bei Variablen mit Referenzdatentyp

Variable mit Referenzdatentyp final

- Referenzdatentyp (Verweis auf Instanz eines Objektes)
- Zuweisung eines Objektes möglich
- Änderung des zuerst zugewiesenen Objektes ist möglich
- keine weitere Zuweisung eines anderen Objektes möglich

```
final MyFooClass instanzMyFooClass = new MyFooClass(1);  
instanzMyFooClass.setEineVariable(5);  
instanzMyFooClass = new MyFooClass(10);
```

Führt zu Compilerfehler:
**“The final local variable
instanzMyFooClass cannot be assigned.
It must be blank and not using a
compound assignment”**

ATTRIBUT ALS FINAL

final bei Attribut


Attribute final

- nur eine Wertzuweisung möglich
 - nach Deklaration
 - im Constructor
 - im Initializer-Block
 - im static Initializer-Block

Bezeichnung bei final


- Snake Case (All Caps)
 - EIN_BESONDERER_VARIABLENNAME

```
public class MyFooClass {  
    private final int FINAL_VAR = 23;  
    ...  
}
```



Zuweisung direkt nach
Deklaration möglich

```
public class MyFooClass {  
    private final int FINAL_VAR;  
    public MyFooClass(int wert) {  
        FINAL_VAR = 23;  
    }  
    ...  
}
```



Zuweisung im
Constructor möglich

ARGUMENT ALS FINAL

final bei Attribut

Attribut final

- Wertzuweisung nur bei der Übergabe des Argumentwertes
- Änderung in Funktion nicht möglich

```
public void methodWithFinalVar(final int finalVar) {  
    finalVar = 10;  
}
```

Führt zu Compilerfehler:
“The final local variable finalVar cannot be assigned. It must be blank and not using a compound assignment”

Flexibilität vermindern mit final

- Verhindern von Änderung oder Erweiterung unter anderem durch **final**
- Wirkung möglich auf
 - Klasse
 - Methoden
 - Variablen
 - Attribute