

Java Kurs 2/13

Florian Pix 2018

Inhalt

1. Wiederholung
2. if, for, while
3. OOP in Java

Wiederholung

- int, long
- float, double
- Char
- String
- `public static void main(String[] args)`
- ;

if

```
if (condition) {  
    //do something if condition is true  
}
```

```
if (a == b) {  
    //if a = b then do something  
}
```

else

```
if (condition) {  
    //do something if condition is true  
} else {  
    //do something if condition is false  
}
```

```
if (a == b) {  
    //do something if a = b  
} else {  
    //do something if a != b  
}
```

else if

```
if ( condition ) {  
    //do something if condition is true  
} else if ( otherCondition ) {  
    //do something if condition is false and otherCondition is true  
} else {  
    //do something if condition is false  
}
```

else if

```
int myNumber = 5;
if ( myNumber == 3) {
    System . out . println (" Won't be printed because myNumber != 3");
} else if ( myNumber == 2) {
    System . out . println (" Won't be printed because myNumber != 2");
} else {
    System . out . println (" Will be printed because myNumber != 3");
}
```

Conditions

- == Gleich
- != Ungleich
- > Größer als
- < Kleiner als
- >= Größer/Gleich
- <= Kleiner/Gleich
- true
- false
- && AND
- || OR

while

```
while ( condition ) {  
    // do this as long as condition is true  
}
```

```
while ( a == b ) {  
    // do this as long a = b  
}
```

○ Denkt an die Abbruchbedingung!

while

```
public class WhileExample {  
    public static void main ( String [] args ) {  
        int a = 0;  
        while (a <= 10) {  
            System . out . println (a) ;  
            a ++; // Otherwise you would get an endless loop  
        }  
    }  
}
```

for

```
for( initialvalue, condition, change) {  
    //do this while condition is true  
}
```

```
for( int a = 0, a <= 10, a++) {  
    //do this while a <= 10  
}
```

for

```
public class ForExample {  
    public static void main ( String [] args ) {  
        for ( int i = 0; i <= 10; i ++ ) {  
            System . out . print ( "na " );  
        }  
        System . out . println ( " BATMAN !" );  
    }  
}
```

Object Oriented Programming

with Java

OOP with Java

```
public class Student {  
    // Attributes  
    private String name ;  
    private int matriculationNumber ;  
    // Methods  
    public void hallo(String name) {  
        System.out.println("Hallo" + name);  
    }  
}
```

OOP with Java

- Wir wissen wie man primitive Datentypen deklariert und ihnen ein Wert zuweist

```
int a;
```

```
a = 42;
```

- Objekterstellung bzw. die Instanziierung einer Klasse funktioniert ähnlich

```
Student Clara = new Student();
```

- Die Variable (**Clara**) nennt man auch Referenz.

Konstruktor

```
public class Student {  
    private String name ;  
  
    Student(String name){  
        this.name = name;  
    }  
  
    public String getName () {  
        return name ;  
    }  
  
    public void setName ( String name ) {  
        this.name = name ;  
    }  
}
```


Methoden auf Instanzen rufen

- Nach der Instanziierung einer Klasse können auf ihren Instanzen Methoden aufgerufen werden
- `referenz.methodenName();`

Methoden auf Instanzen rufen

```
public class Student {  
    private String name ;  
  
    Student(String name){  
        this.name = name;  
    }  
  
    public String getName () {  
        return name ;  
    }  
  
    public void setName ( String name ) {  
        this.name = name ;  
    }  
}
```

Methoden auf Instanzen rufen

- `Student clara = new Student("Clara");`

- `clara.getName();` //returns name attribute of clara ("Clara")

Methoden mit Rückgabewerten und Argumenten

```
public class Student {  
    private String name ;  
  
    Student(String name){  
        this.name = name;  
    }  
  
    public String getName () {  
        return name ;  
    }  
  
    public void setName ( String name ) {  
        this.name = name ;  
    }  
}
```

Demo

Du möchtest ein Einschreibungssystem für einen Programmierkurs schreiben.

Es gibt:

Studenten welche an Kursen teilnehmen wollen,
die von Tutoren gehalten werden.

Die Kurse finden in Räumen statt.

Aufgabe

- http://fsr.github.io/java-lessons/exercises/02_library_part_1.html