

Methoden

((itanius informatik))

Java Foundation Track by Carsten Bokeloh

Methoden

- Methoden
- Aufruf von Methoden
- Regeln zum Benennen von Methoden
- Methoden Rückgabewert
- Parameter übergeben
- Methoden überladen
- Variable Argumentanzahl
- main Methode
- Methode aufrufen

Methoden

```
public class Auto {  
    String marke;  
    String farbe;  
    int geschwindigkeit;  
    int baujahr;
```

Rückgabetyp

```
    public void anlassen() {  
        //Code zum Anlassen des Autos  
    }
```

Parameter

```
    public void setGeschwindigkeit (int geschwindigkeit) {  
        this.geschwindigkeit = geschwindigkeit;  
    }
```

```
    public int getGeschwindigkeit() {  
        return this.geschwindigkeit;  
    }
```

Return-Value

Modifier

Methoden

- ❖ Eigenschaften
 - ❖ Besitzen feste Anzahl von Parametern mit jeweils festem Typ
 - ❖ Besitzen immer einen Rückgabotyp
 - ❖ Gehören immer zu Klassen oder Objekten
 - ❖ (Es gibt keine globalen Funktionen)

Aufruf von Methoden

- ❖ Methoden werden innerhalb der eigenen Klasse direkt über ihren Namen mit () aufgerufen.
 - ❖ `int i = getAge();`
- ❖ Außerhalb der Klasse, über die Objekt Referenz gefolgt von einem Punkt und dann der Name mit ().
 - ❖ `Man adam = new Man();`
 - ❖ `int age = adam.getAge();`

Methoden aufrufen

- ❖ weglassen von static erzeugt Instanzmethoden
- ❖ nur mit Objekt der Klasse verfügbar.

```
public class Summierer {  
    public int summiere(int x, int y) {  
        return x + y;  
    }  
}
```

```
Summierer summierer = new Summierer();  
summierer.summiere(4, 5);
```

Regeln zum Benennen von Methoden

- ❖ Der Name der Methode sollte deren **Aufgabe** beschreiben.
- ❖ Es sollte **immer** ein Verb sein.
- ❖ Der Name der Methode sollte immer mit einem kleinen Buchstaben beginnen
- ❖ Danach sollte jedes neue Wort mit einem Großbuchstaben beginnen

Parameterübergabe

- ❖ Parameter werden immer **byValue** übergeben.
- ❖ Bei primitiven Datentypen wird eine Kopie der Wertes übergeben.
- ❖ Bei Referenz-Datentypen wird eine Kopie der Referenz übergeben.

**!!!Rosen sind rot,
dieser Vers ist für Sie,
übergeb ich nen Wert
ist das ne Kopie!!!**

Parameterübergabe mit primitiven Datentypen

```
public void aendern(int i) {  
    i = 20;  
}
```

```
public void test() {  
    int i, j;  
    i = 10;  
    aendern(i);  
    j = i; // i = 10;  
    System.out.println(i);  
}
```


Parameterübergabe mit Referenzdatentypen

```
Man man = new Man();  
man.setAge(10);  
aendern(man);  
int age = man.getAge();  
System.out.println(age);
```

```
void aendern(Man man) {  
    man.setAge(20);  
}
```


Methoden überladen

möglich durch:

Eine Methode heißt gleich, akzeptiert aber eine unterschiedliche Anzahl von Argumenten

Eine Methode heißt gleich, hat aber für den Compiler unterschiedliche Parametertypen

Variable Argument-Anzahl bei Methoden

```
public static int summiere(int... werte) {  
    int summe = 0;  
    for (int x : werte)  
        summe = summe + x;  
    return summe;  
}
```

```
summiere(3, 4, 5, 6);
```

```
summiere(7, 8, 9, 10, 11, 12);
```


Übung

Schreiben Sie eine Methode `max`, die das Maximum zweier `int` Zahlen, die an die Methode übergeben werden, ermittelt. Sie können zur Ermittlung die Methode Java `java.lang.Math.max` benutzen.

Schreiben Sie in der gleichen Klasse eine zweite Methode `max`, die das Maximum zweier ihr zu übergebenden `double` Werte ermittelt. Auch hier dürfen Sie sich der Klasse `java.lang.Math.max` bedienen.

Rufen Sie in der `main` Methode beide Methoden auf und geben Sie das Ergebnis über `System.out.println` aus.

main-Methode

- ❖ `public static void main(String arg[])`
- ❖ was steht im String array?

```
public class Hello {  
    public static void main(String args[]) {  
        System.out.println("Hallo " + args[0] + " " + args[1]);  
    }  
}
```


Methodenaufrufe

- ❖ static bedeutet wie bei den Variablen auch Klassenmethoden
- ❖ `public static void summiere(int x, int y)`
- ❖ können benutzt werden über statischen import
 - ❖ `import static <<Paketname>>.<<Klassenname>>.<<Komponentenname>>;`
 - ❖ `import static <<Paketname>>.<<Klassenname>>.*;`
 - ❖ `import static java.lang.Math.*;`

Im Buch gibt es mehr Infos unter Kapitel 2.7

[http://openbook.galileocomputing.de/javainsel/
javainsel_02_007.html#dodtp0e6fae8f-4221-41a2-9467-2728f1eff1c7](http://openbook.galileocomputing.de/javainsel/javainsel_02_007.html#dodtp0e6fae8f-4221-41a2-9467-2728f1eff1c7)

Übung

Nehmen Sie einige Ihrer vorigen Übungsprogramme zur Hand und schreiben Sie ein kurzes Menü, mit dem Sie diese starten können. Verwenden Sie hierzu die IOtools, um eine Zahl zwischen eins und drei eingeben zu lassen. Starten Sie bei der Zahl 1 z.B. das Sortierprogramm und bei 2 und drei je ein weiteres Programm Ihrer Wahl.

Hinweis. Das Sortierprogramm können Sie starten mit `Sortierprogramm.main(args)`, wobei `args` ein beliebiges Feld von Zeichenketten ist (beispielsweise das Feld `args`, das der Hauptmethode Ihres neuen Menüs übergeben wurde).