

# Methoden in Java

# Syntax

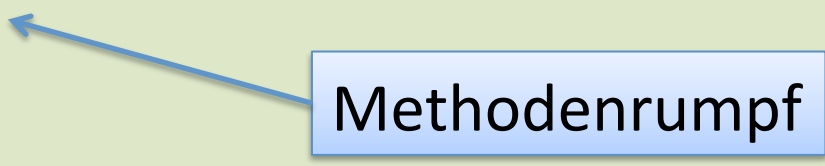
```
public class Kunde
{
    // Konstruktor
    public Kunde()
    {
    }

    // Methoden
    public void geldAbheben()
    {
    }
}
```

# Syntax

```
public class Kunde
{
    // Konstruktor
    public Kunde()
    {
    }

    // Methoden
    public void geldAbheben()
    {
        // Anweisungen ("Statements")
    }
}
```



Methodenrumpf

```
public void geldAbheben ()
```

Zugriffsmodifikator  
(public, protected, private)

Rückgabety

Name der Methode  
(Bezeichner)

"Signatur" der Methode

# Parameterliste

```
public void groesseAngeben(int x, int y)
```

Parameterliste;  
Wenn keine Parameter vorhanden, wird eine  
leere Parameterliste angegeben:

```
public void fehlermeldungAusgeben ()
```

# Parameterliste

```
public void groesseAngeben(int x, int y)
```

Parameterliste;

Wenn keine Parameter vorhanden, wird eine leere Parameterliste angegeben:

**ACHTUNG: Unterschied zum Klassendiagramm!**

**Dort**

```
groesseAngeben(x:int, y:int)
```

# Rückgabetyt – Beispiel 1

```
public void etwasSagen()  
{  
    System.out.println („Hallo.“) ;  
}
```

Methode schreibt nichts in den Speicher,  
Computer muss sich nichts merken.

Ergebnis = „void“ (leer, nichts)

## Rückgabebetyp – Beispiel 2

```
public int addieren(int a, int b)
{
    int c = a + b;
    return c;
}
```

Computer „merkt“ sich den Wert von c.  
Andere Methoden können mit c arbeiten.

Ergebnis der Methode = „int“



# Rückgabebetyp – Beispiel 2

```
public int addieren(int a, int b)
{
    int c = a + b;
    return c;
}
```

**Datentyp angeben!**

# Übung Rückgabetyt

**Nennen Sie einen möglichen Rückgabewert!**

Methode	möglicher Rückgabewert
<code>public int berechneNeuenKontostand()</code>	350
<code>public double berechneRadius()</code>	
<code>public void fehlermeldungAusgeben()</code>	
<code>public void berechneRadius()</code>	
<code>public String fehlermeldungErzeugen()</code>	
<code>public boolean kundenkontoPruefen()</code>	