



# Anweisungen, Ausdrücke und Operatoren



## Anweisungen und Ausdrücke

### Operatoren

Mathematische Operatoren

Zusammengesetzte Operatoren

Inkrementieren/Dekrementieren

Operatorpräzedenz

Vergleichsoperatoren

Übersicht

### Die If-Anweisung

Die Else-Klausel

### Max

### Logische Operatoren

### Rückblick: String-Vergleich

### Präzedenztabelle für Java

### Fragerunde



- ▶ Whitespaces (Tab, Leerzeichen) werden ignoriert
- ▶ Anweisungen die Werte zurückgeben nennt man **Ausdrücke**

```
int i = 1;  
System.out.println("Der Dozent  
ist toll");  
int counter = 5;
```

## Erinnerung

Jede Java Anweisung muss mit einem ; beendet werden



- ▶ Symbol das den **Compiler** zur Ausführung einer Aktion veranlasst
- ▶ Zuweisungsoperator (=) schon bekannt
- ▶ Java kennt:
  - ▶ Mathematische Operatoren
  - ▶ Vergleichsoperatoren
  - ▶ Bitoperatoren

# Operatoren

## Mathematische Operatoren

- ▶ Addition: +
- ▶ Subtraktion: -
- ▶ Multiplikation: \*
- ▶ Division: /
- ▶ Modulo: %

### Hinweis

- ▶ Zweite Klasse Mathematik
- ▶  $23/4 = 5 \text{ Rest } 3$
- ▶ Modulo liefert den Rest einer Division
- ▶  $23\%4 = 3$

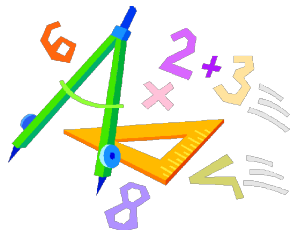


Abbildung : Math [1]

# Operatoren

## Zusammengesetzte Operatoren



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Häufig werden Variablen geändert und der Wert in der Variable gespeichert

```
meinAlter = meinAlter + 2;  
meinAlter += 2; // Plus-Gleich
```

- ▶ Es gibt noch weitere:

- ▶ -=
- ▶ \*=
- ▶ /=
- ▶ %=

# Operatoren

## Inkrementieren/Dekrementieren



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Der Operator **++** erhöht den Wert einer Variable genau um den Wert 1
- ▶ **a++**; gleich mit **a+=1** gleich mit **a = a + 1**;
- ▶ In seiner Verwendung jedoch, hat er unterschiedliche Auswirkungen:

### ▶ Präfix

```
i = 0;  
System.out.println(i); // 0  
System.out.println(++i); // 1  
System.out.println(i); // 1
```

### ▶ Postfix

```
i = 0;  
System.out.println(i); // 0  
System.out.println(i++); // 0  
System.out.println(i); // 1
```

## Hinweis

Analog mit dem Dekrementoperator **--**

# Operatoren

## Operatorpräzedenz



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

$$y = 6 + 4 / 2$$

Sekunden = minutenNachdenken + minutenTippen \* 60;

Sekunden = (minutenNachdenken + minutenTippen) \* 60;

- ▶ Die Operatorpräzedenz gibt die Reihenfolge vor, in der Ausdrücke ausgewertet werden
- ▶ Inkremente und Dekremente, arithmetische Ausdrücke, vergleiche, logische Ausdrücke
- ▶ Zuweisungen werden zuletzt ausgewertet
- ▶ Punkt vor Strich :)
- ▶ Klammern können Reihenfolge ändern



# Operatoren

## Vergleichsoperatoren



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Vergleichsoperatoren liefern einen Rückgabewert vom Typ boolean
- ▶ Beispiel

```
meinAlter == deinAlter //false  
meinAlter > deinAlter //true  
  
//Was liefert?  
System.out.println( 1 < 2 );
```

# Operatoren

## Übersicht



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Operator	Symbol	Beispiel	Rückgabewert
Gleich	==	100 == 50;	FALSE
		50 == 50;	TRUE
Ungleich	!=	100 != 50;	TRUE
		50 != 50	FALSE
Größer als	>	100 > 50;	TRUE
		50 > 50;	FALSE
Größer als oder Gleich	>=	100 >= 50;	TRUE
		50 >= 50	TRUE
Kleiner als	<	100 < 50;	FALSE
		50 < 50;	FALSE
Kleiner als oder Gleich	<=	100 <= 50;	FALSE
		50 <= 50;	TRUE

# Die If-Anweisung



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Normalerweise verläuft ein Programm sequentiell d.h zeilenweise
- ▶ If-Anweisungen testen Bedingungen

```
If (Ausdruck)  
    Anweisung ;
```

- ▶ Bei **false** wird der Block übersprungen bei **true** wird dieser ausgeführt

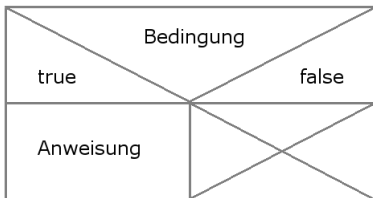


Abbildung : Struktogram

# Die If-Anweisung

## Beispiel



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

```
int meinAlter = 20;

//Darf ich Alkohol in den USA trinken?
if (meinAlter >= 21)
    System.out.print("ja");

//Geburtstag
meinAlter++;

//Darf ich jetzt Alkohol in den USA trinken?
if (meinAlter >= 21)
    System.out.print("ja");
```

## Hinweis

Achtung hinter If-Anweisungen kein ;

# Die If-Anweisung

## Die Else-Klausel



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- Nach **if**: das **else** es bezeichnet einen alternativ auszuführenden Teil des Codes

```
int meinAlter = 20;

// Darf ich Alkohol in den USA trinken?
if (meinAlter >= 21) {
    System.out.print("ja");
    System.out.print("Prost!");
}
else
    System.out.print("nein");
```

## Hinweis

Einzeilige Anweisungen benötigen keine Klammern



```
int a = 2, b = 1;

// Part 1
if(a > b)
    System.out.println("Variable a is bigger with value:" + a);

// Part 2
// Note: Equals case -> a = b
int max = 0;
if(a > b)
    max = a;
else
    max = b;

// Part 3
int max = a > b ? a : b;
```



- ▶ logisches **Und** (&&)  $\Rightarrow$  (Ausdruck1 && Ausdruck2)
- ▶ logisches **Oder** (||)  $\Rightarrow$  (Ausdruck1 || Ausdruck2)
- ▶ logisches **Nicht** (!)  $\Rightarrow$  !Ausdruck1

## Suche Belegung: True oder False?

```
if ((x==7) && (y==5))  
if ((x==4) || (y>9))  
if (!(x==3))
```

## Hinweis

Die Rangfolge der Operatoren ist auch hier zu beachten. Ggf. Klammern setzen.



## ► Strings vergleichen

```
if (string1 == string2) {  
    System.out.println("String1 und String2 liegen an der gleichen  
        Stelle im Speicher");  
}  
  
if (string1.equals(string2)) {  
    System.out.println("String1 und String2 haben den gleichen  
        Inhalt");  
}
```

## Fragerunde

```
String string1 = "Pizza"; if (string1 == string2)  
String string2 = "Pizza"; if (string1.equals(string2))
```





<b>Prio 1</b>	++, - -
<b>Prio 2</b>	*, /, %
<b>Prio 3</b>	+, - und + für String-Konkatenation
<b>Prio 4</b>	<<, >>, <<<, >>>
<b>Prio 5</b>	<, <=, >, >= und instanceof
<b>Prio 6</b>	==, !=
<b>Prio 7-11</b>	&, ^,  , &&,
<b>Prio 12</b>	?:
<b>Prio 13</b>	= sowie *=, /=, %=, +=, -=

## Hinweis

Die geringste Priorität bindet dabei am stärksten



- ▶ Welche Wirkung haben Tabulator, Leerzeichen und Zeilenumbrüche auf ein Programm?
- ▶ Was ist der Wert von  $201/4$  (Integer)?
- ▶ Was ist der Wert von  $201 \% 4$ ?
- ▶ Was ist der Unterschied zwischen `if(x = 3)` und `if(x == 3)`?



- ▶ **[1]** [http://www.aaps.k12.mi.us/ins.home/ins.esl/elementary\\_math\\_vocabulary](http://www.aaps.k12.mi.us/ins.home/ins.esl/elementary_math_vocabulary)