

Leonard Berresheim

03 Operatoren

M21 - Grundlagen der Programmierung

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität
+	positives Vorzeichen	1
-	negatives Vorzeichen	1
++	Inkrementierung	1
--	Dekrementierung	1
*	Multiplikation	2
/	Division	2
%	Rest	2
+	Addition	3
-	Subtraktion	3

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität	int zahlA = 10; int zahlB;	Wert von zahlB
+	positives Vorzeichen	1	zahlB = + zahlA;	(+)10
-	negatives Vorzeichen	1	zahlB = - zahlA;	-10
++	Inkrementierung	1		
--	Dekrementierung	1		
*	Multiplikation	2	zahlB = zahlA * 3;	30
/	Division	2	zahlB = zahlA / 2;	5
%	Rest	2	zahlB = zahlA % 3;	1
+	Addition	3	zahlB = ZahlA + 10;	20
-	Subtraktion	3	zahlB = ZahlA - 10;	0

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität
+	positives Vorzeichen	1
-	negatives Vorzeichen	1
++	Inkrementierung	1
--	Dekrementierung	1
*	Multiplikation	2
/	Division	2
%	Rest	2
+	Addition	3
-	Subtraktion	3

```
zahlA = 10;  
zahlA++; //zahlA hat den Wert 11  
zahlA--; //zahlA hat den Wert 10
```

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität
+	positives Vorzeichen	1
-	negatives Vorzeichen	1
++	Inkrementierung	1
--	Dekrementierung	1
*	Multiplikation	2
/	Division	2
%	Rest	2
+	Addition	3
-	Subtraktion	3

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 10;
```

```
zahlA = -zahlA / 2 + 2;
```

zahlA == ?

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität
+	positives Vorzeichen	1
-	negatives Vorzeichen	1
++	Inkrementierung	1
--	Dekrementierung	1
*	Multiplikation	2
/	Division	2
%	Rest	2
+	Addition	3
-	Subtraktion	3

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 10;
```

```
zahlA = -zahlA / 2 + 2;
```

```
zahlA = (((-zahlA) / 2) + 2);
```

zahlA == -3

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität
+	positives Vorzeichen	1
-	negatives Vorzeichen	1
++	Inkrementierung	1
--	Dekrementierung	1
*	Multiplikation	2
/	Division	2
%	Rest	2
+	Addition	3
-	Subtraktion	3

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 10;
```

```
zahlA = ++zahlA % 2 - 2;
```

zahlA == ?

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität
+	positives Vorzeichen	1
-	negatives Vorzeichen	1
++	Inkrementierung	1
--	Dekrementierung	1
*	Multiplikation	2
/	Division	2
%	Rest	2
+	Addition	3
-	Subtraktion	3

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 10;
```

```
zahlA = ++zahlA % 2 - 2;
```

```
zahlA = (((++zahlA) % 2) - 2);
```

zahlA == -1

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität
+	positives Vorzeichen	1
-	negatives Vorzeichen	1
++	Inkrementierung	1
--	Dekrementierung	1
*	Multiplikation	2
/	Division	2
%	Rest	2
+	Addition	3
-	Subtraktion	3

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 10;
```

```
zahlA = -(-zahlA++ / (2 + 2) * 5);
```

zahlA == ?

Operatoren

Arithmetische Operatoren

Operator	Bezeichnung	Priorität
+	positives Vorzeichen	1
-	negatives Vorzeichen	1
++	Inkrementierung	1
--	Dekrementierung	1
*	Multiplikation	2
/	Division	2
%	Rest	2
+	Addition	3
-	Subtraktion	3

Arithmetische Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 10;
```

```
zahlA = -(-zahlA++ / (2 + 2) * 5);
```

```
zahlA = -((( -zahlA++ ) / (2 + 2)) * 4);
```

zahlA == 4

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

Vergleich zweier Elemente.

Liefert als Ergebnis ein **Wahrheitswert** (boolean).

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == ?

```
isTrue = zahlA < 10
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == true

```
isTrue = zahlA < 10
```

```
isTrue = 9 < 10
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == ?

```
isTrue = zahlA > 9
```

```
isTrue = 9 > 9
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == false

```
isTrue = zahlA > 9
```

```
isTrue = 9 > 9
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == ?

```
isTrue = zahlA >= 9
```


Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == true

```
isTrue = zahlA >= 9
```

```
isTrue = 9 >= 9
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == ?

```
isTrue = zahlA == 8
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == ?

```
isTrue = zahlA == 8
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == true

```
isTrue = zahlA != 8
```

```
isTrue = 9 != 8
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == ?

```
isTrue = zahlA != 8
```

Operatoren

Relationale Operatoren

Operator	Bezeichnung	Priorität
<	kleiner	5
<=	kleiner oder gleich	5
>	größer	5
>=	größer oder gleich	5
==	gleich	6
!=	ungleich	6

Relationale Operatoren - Grundkurs Java von Dietmar Abts

```
int zahlA = 9;
```

```
boolean isTrue;
```

isTrue == true

```
isTrue = zahlA != 8
```

```
isTrue = 9 != 8
```

Operatoren

Logische Operatoren

Operator	Bezeichnung	Priorität
!	NICHT	1
&	UND mit vollständiger Auswertung	7
^	exklusives ODER (XOR)	8
	ODER mit vollständiger Auswertung	9
&&	UND mit kurzer Auswertung	10
	ODER mit kurzer Auswertung	11

Logische Operatoren - Grundkurs Java von Dietmar Abts

Zur **Durchführung** logischer Operationen.

Berechnung mit **Wahrheitswerten**.

Operatoren

Logische Operatoren

Operator	Bezeichnung	Priorität
!	NICHT	1
&	UND mit vollständiger Auswertung	7
^	exklusives ODER (XOR)	8
	ODER mit vollständiger Auswertung	9
&&	UND mit kurzer Auswertung	10
	ODER mit kurzer Auswertung	11

Logische Operatoren - Grundkurs Java von Dietmar Abts

```
boolean isTrue = true;
```

isTrue == ?

```
isTrue = !isTrue
```


Operatoren

Logische Operatoren

Operator	Bezeichnung	Priorität
!	NICHT	1
&	UND mit vollständiger Auswertung	7
^	exklusives ODER (XOR)	8
	ODER mit vollständiger Auswertung	9
&&	UND mit kurzer Auswertung	10
	ODER mit kurzer Auswertung	11

Logische Operatoren - Grundkurs Java von Dietmar Abts

```
bool isTrue = true;
```

isTrue == false

```
isTrue = !isTrue
```

```
isTrue = !true
```

Logische Operatoren

& - Wahrheitstabelle

A	B	A & B
false	false	false
false	true	false
true	false	false
true	true	true

```
boolean A = true;
```

```
boolean B = false
```

```
A = A & B
```

A == ?

Logische Operatoren

& - Wahrheitstabelle

A	B	A & B
false	false	false
false	true	false
true	false	false
true	true	true

```
boolean A = true;
```

```
boolean B = false
```

```
A = A & B
```

```
A = true & false
```

A == false

Logische Operatoren

|| - Wahrheitstabelle

A	B	A B
false	false	false
false	true	true
true	false	true
true	true	true

```
boolean A = true;
```

```
boolean B = false
```

```
A = A || B
```

A == ?

Logische Operatoren

|| - Wahrheitstabelle

A	B	A B
false	false	false
false	true	true
true	false	true
true	true	true

```
boolean A = true;
```

```
boolean B = false
```

```
A = A || B
```

```
A = true || false
```

A == true

Logische Operatoren

XOR - Wahrheitstabelle

XOR - *Entweder oder*

A	B	$A \wedge B$
false	false	false
false	true	true
true	false	true
true	true	false

```
boolean A = true;
```

```
boolean B = false
```

```
A = A ^ B
```

A == ?

Logische Operatoren

XOR - Wahrheitstabelle

XOR - Entweder oder

A	B	$A \wedge B$
false	false	false
false	true	true
true	false	true
true	true	false

```
boolean A = true;
```

```
boolean B = false
```

```
A = A ^ B
```

```
A = true ^ false
```

A == true

Operatoren

Kombination

```
int zahlA = 10;
```

```
int zahlB = 5;
```

```
bool isTrue;
```

```
isTrue = (zahlA > zahlB) == true;
```

```
zahlA > zahlB -> 10 > 5 -> true
```

isTrue == ?

Operatoren

Kombination

```
int zahlA = 10;
```

```
int zahlB = 5;
```

```
bool isTrue;
```

```
isTrue = (zahlA > zahlB) == true;
```

```
zahlA > zahlB -> 10 > 5 -> true
```

```
isTrue = true == true;
```

*isTrue == **true***

Operatoren

Typumwandlung

Was passiert wenn **Variablen** von unterschiedlichem **Datentyp** miteinander verrechnet werden?

Beispiel:

```
int ganzZahl = 5;
```

```
float gleitKommaZahl = 4.5;
```

```
ganzZahl + gleitKommaZahl
```

Operatoren

Typumwandlung

Was passiert wenn **Variablen** von unterschiedlichem **Datentyp** miteinander verrechnet werden?

Beispiel:

```
int ganzZahl = 5;  
float gleitKommaZahl = 4.5;
```

`ganzZahl + gleitKommaZahl`

Der “kleinere” Datentyp wird (vom compiler) **umgewandelt** in den “größeren”. Hier zum Beispiel wird der **int** in ein **float** umgewandelt.

Die **Umwandlung** ist notwendig, da nur Variablen vom gleichen **Datentyp** miteinander verrechnet werden können.

Operatoren

Typumwandlung

Was passiert wenn das **Ergebnis** einer **Berechnung** nicht dem ursprünglichen **Datentyp** entspricht.

Beispiel:

```
int ganzZahlA = 5;
```

```
int ganzZahlB = 2;
```

```
int ergebnis = ganzZahlA / ganzZahlB;
```

Operatoren

Typumwandlung

Was passiert wenn das **Ergebnis** einer **Berechnung** nicht dem ursprünglichen **Datentyp** entspricht.

Beispiel:

```
public class Main {  
  
    public static void main(String[] args) {  
        int ganzZahl = 5;  
        float gleitKommaZahl = 4.5f;  
        int ergebnis = ganzZahl + gleitKommaZahl;  
    }  
}
```

Required type: int
Provided: float
Cast to 'int' Alt+Shift+Enter More actions... Alt+Enter
float gleitKommaZahl = 4.5f

Operatoren

Typumwandlung

Was passiert wenn das **Ergebnis** einer **Berechnung** nicht dem ursprünglichen **Datentyp** entspricht.

Zwei mögliche Lösungen!

Lösung 1:

```
int ganzZahlA = 5;
```

```
int ganzZahlB = 2;
```

```
float ergebnis = ganzZahlA / ganzZahlB;
```

Operatoren

Cast-Operator

Was aber wenn das Ergebnis ein **int** sein soll?

Lösung 2:

Cast Operator

Schreibweise:

`(datentyp) variable`

Operatoren

Cast-Operator

Was aber wenn das Ergebnis ein **int** sein soll?

Lösung 2:

Cast Operator

Schreibweise:

`(datentyp) variable`

Beispiel:

```
int ganzZahlA = 5;
```

```
int ganzZahlB = 2;
```

```
int ergebnis = (int) (ganzZahlA / ganzZahlB);
```


Operatoren

Cast-Operator

Achtung!

```
int ergebnis = (int) (ganzZahlA / ganzZahlB);
```



!=

```
int ergebnis = (int) ganzZahlA / ganzZahlB;
```



=

```
int ergebnis = ((int) ganzZahlA) / ganzZahlB;
```





www.htw-berlin.de