

Software Development

Operatoren

Software-Lösungen. Einfach. Clever.



Inhalt

Operatoren verknüpfen Variablen bzw. Operanden miteinander und führen Berechnungen durch.

- Arithmetische Operatoren
- Zuweisungsoperatoren
- Logische Operatoren
- Vergleichsoperatoren

Arithmetische Operatoren

Software-Lösungen. Einfach. Clever.

Arithmetische Operatoren

Standard-Operatoren

Operator	Beispielausdruck	Erklärung
+	$x + y$	Addition
-	$x - y$	Subtraktion
*	$x * y$	Multiplikation
/	x / y	Division
%	$x \% y$	Modulo-Division (liefert Restwert)

Arithmetische Operatoren

Standard-Operatoren

C #

```
int a = 10 + 20;  
Console.WriteLine(a);  
  
int b = 30 - 10;  
Console.WriteLine(b);  
  
int c = 5 * 4;  
Console.WriteLine(c);  
  
int d = 50 / 10;  
Console.WriteLine(d);  
  
int e = 50 % 7;  
Console.WriteLine(e);
```

Terminal – Beispiele

```
30  
20  
20  
5  
1  
█
```

Arithmetische Operatoren

Inkrement- und Dekrement-Operatoren

Operator	Beispielausdruck	Erklärung
++	$x++$ $++x$	Postfix-Inkrement Präfix-Inkrement
--	$x--$ $--x$	Postfix-Dekrement Präfix-Dekrement

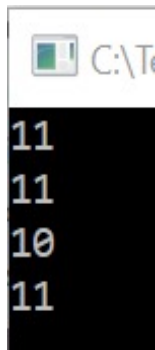
Arithmetische Operatoren

Inkrement- und Dekrement-Operatoren

C #

```
int a = 10;  
Console.WriteLine(++a);  
Console.WriteLine(a);  
  
int b = 10;  
Console.WriteLine(b++);  
Console.WriteLine(b);
```

Output



```
11  
11  
10  
11
```


Zuweisungsoperatoren

Software-Lösungen. Einfach. Clever.

Zuweisungsoperatoren

Operator	Beispielausdruck	Erklärung
=	$x = y$	x wird der Wert von y zugewiesen
+=	$x += y$	x ergibt sich zu $x + y$
-=	$x -= y$	x ergibt sich zu $x - y$
*=	$x *= y$	x ergibt sich zu $x * y$
/=	$x /= y$	x ergibt sich zu x / y
%=	$x \% = y$	x ergibt sich als Restbetrag aus x / y

Logische Operatoren

Software-Lösungen. Einfach. Clever.

Logische Operatoren

Vergleichsoperatoren

Operator	Erklärung
==	x gleich y ?
!=	x ungleich y ?
<	x kleiner y ?
<=	x kleiner oder gleich y ?
>	x grösser y ?
>=	x grösser oder gleich y ?

Logische Operatoren

Boolesche Operatoren

Operator	Erklärung
&	Und: Liefert <i>true</i> , wenn beide Operanden <i>true</i> sind
	Oder: Liefert <i>true</i> , wenn mindestens einer der Operanden <i>true</i> ist
^	Exklusiv-Oder (XOR): Liefert <i>true</i> , wenn genau nur einer der beiden Operanden <i>true</i> ist
&&	intelligentes Und: wie &-Operator; ist aber der erste Operand <i>false</i> , wird der zweite nicht ausgewertet
	intelligentes Oder: wie -Operator; ist aber der erste Operand <i>true</i> , wird der zweite nicht ausgewertet.
!	Negation: Aus <i>true</i> wird <i>false</i> und aus <i>false</i> wird <i>true</i>

Logische Operatoren

Bitweise Operationen

Operator	Erklärung
&	bitweise "UND"-Verknüpfung der beiden Operanden
	bitweise "ODER"-Verknüpfung der beiden Operanden
^	bitweise "XOR"-Verknüpfung der beiden Operanden
>>	Rechtsverschiebung aller Bits eines Operanden um eine bestimmte Anzahl
<<	Linksverschiebung aller Bits eines Operanden um eine bestimmte Anzahl

C

```
int a = 1;    // Bitmuster = 001
int b = 7;    // Bitmuster = 111
a = a ^ b;    // Bitmuster = 110 -> a erhält den Wert 6
```

C

```
int a = 1;    // Bitmuster = 001
a = a << 2;    // Bitmuster = 100 -> a erhält den Wert 4
```

Rangfolge der Operatoren

Operator	Bedeutung
()	Klammern
!	logisches NOT
* / %	Multiplikation, Division, Modulo
+-	Addition, Subtraktion
< <= > >=	kleiner als, kleiner gleich als, grösser als, grösser gleich als
== !=	gleich, ungleich

Beispiel Arithmetische Operationen

C #

```
double x = 2.0;  
double y = x * x + 1 + x / 4           // y = 5.5
```

aber

```
double y = x * (x + 1 + x / 4);       // y = 7
```

Beispiel Boolesche Operationen

C #

```
bool b = !true && false || 5 > 6;    // b = false  
int z = 50;  
bool numeric = z > 47 && z < 58;      // numeric = true
```

Übungen

Übungen\Kurseinheit 3/ULogischeOperatoren

Übungen\Kurseinheit 3/UOperatoren



Software-Lösungen. Einfach. Clever.