

# C#

## Überladen von Operatoren

# überladbare Operatoren

## unäre Operatoren:

+	-	// Vorzeichen
!	~	// log. Negation und bitweises Komplement
++	--	// Inkrement und Dekrement

## binäre Operatoren:

+	-	*	/	%	// Arithmetik
==	!=	<	<=	>=	> // Vergleich
&	!	^			// Logik (and, or, xor)
<<	>>				// Verschiebung

## Konvertierungsoperatoren:

implicit operator Typ (Typ arg)

explicit operator Typ (Typ arg)

[]-Operator kann in C# nicht überladen werden.

(=> Indexer)

# Regeln für die Überladung

- Operatoren müssen public und static sein
- Operatorparameter müssen Wertparameter sein (keine ref- oder out-Parameter)
- Unäre und Konvertierungs-Operatoren haben einen Parameter
- Binäre Operatoren haben zwei Parameter
- Folgende Operatoren dürfen nur zusammen überladen werden:

(true false), (== !=), (< >), (<= >=)

# Regeln für die Überladung

- Mit (`==` `!=`) müssen auch `Equals` und `GetHashCode` überschrieben werden, so dass `Collections` und `Hashtabellen` zuverlässig mit dem Typ arbeiten können.
- Beim Überladen von (`<` `>`) und (`<=` `>=`) müssen auch `Comparable` und `Comparable<T>` implementiert werden.

# Beispiel

```
class Demo {  
    int value;  
    public Demo(int value){  
        this.value = value;  
    }  
    public static Demo operator +(Demo a, Demo b){  
        return new Demo(a.value + b.value);  
    }  
    public static Demo operator -(Demo a, int b){  
        return new Demo(a.value - b);  
    }  
    static void Main(string[] args){  
        Demo obj1 = new Demo(523);  
        Demo obj2 = new Demo(4);  
        Demo obj3 = obj1 + obj2;}  
    ...  
}
```