ZUSAMMENFASSUNG Rechenoperatoren

EINFACHE RECHENOPERATIONEN (THEORIE)

- In C# können Rechenoperationen auf alle numerischen Werte angewendet werden
- Dazu verwendet man sogenannte Operatoren (z. B. +, -, *, /)

Einen Überblick über die einfachen Rechenoperatoren sehen wir in Abbildung 1:

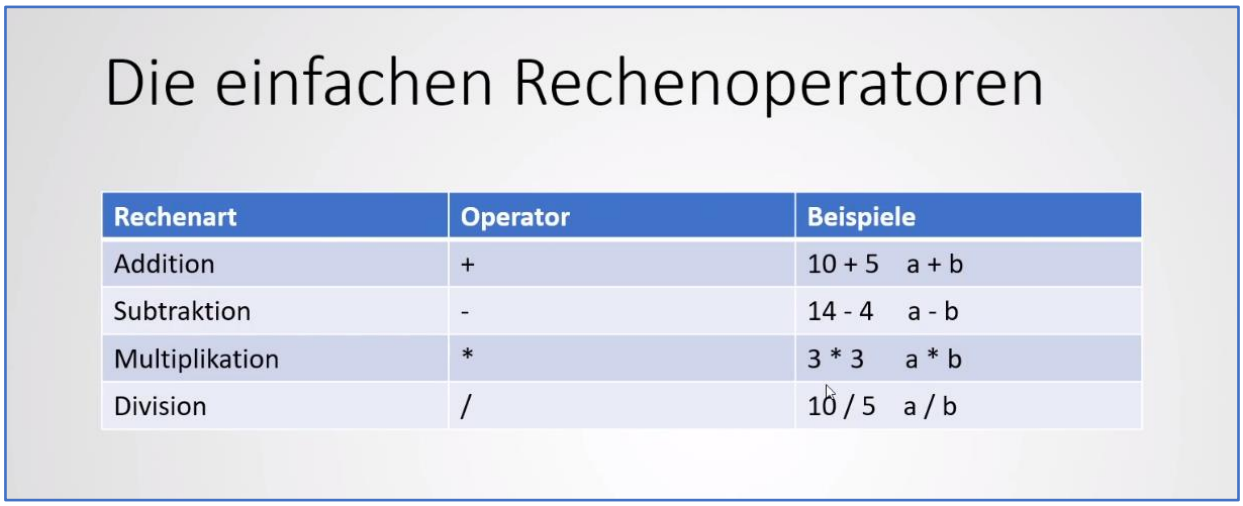

Abbildung 1, einfache Rechenoperatoren

EINFACHE RECHENOPERATIONEN (PRAXIS)

In unserer Entwicklungsumgebung sehen einfache Rechenoperationen folgendermaßen aus:

```
 9    class Program
10    {
11        static void Main(string[] args)
12        {
13            int zahl = 15 + 5;
14
15            Console.WriteLine(zahl);
16            Console.ReadKey();
17        }
18    }
```

Abbildung 2, Rechenoperationen in der Praxis

Es besteht auch die Möglichkeit, Variablennamen stellvertretend für ihre Werte zum Rechnen zu verwenden (s. Abbildungen 3, 4 und 5):

```
 9    class Program
10    {
11        static void Main(string[] args)
12        {
13            int zahl1 = 15;
14            int zahl2 = 5;
15            int ergebnis = zahl1 + zahl2;
16
17            Console.WriteLine(ergebnis);
18            Console.ReadKey();
19        }
20    }
```

Abbildung 3, Addition mit Variablen

```
 9    class Program
10    {
11        static void Main(string[] args)
12        {
13            int zahl1 = 3;
14            int zahl2 = 2;
15            int ergebnis = zahl1 * zahl2;
16
17            Console.WriteLine(ergebnis);
18            Console.ReadKey();
19        }
20    }
```

Abbildung 4, Multiplikation mit Variablen

```
 9    class Program
10    {
11        static void Main(string[] args)
12        {
13            int zahl1 = 10;
14            int zahl2 = 5;
15            int ergebnis = zahl1 / zahl2;
16
17            Console.WriteLine(ergebnis);
18            Console.ReadKey();
19        }
20    }
```

Abbildung 5, Division mit Variablen

Wichtig:

Beim Rechnen gilt auch im Code die Punkt- vor Strichregel:
Beispiel: 10 * 5 + 2 **Ergebnis:** 52

Beeinflussen kann man diese Regel durch das Setzen von Klammern:
Beispiel: 10 * (5 + 2) **Ergebnis:** 70

Um den Wert einer Variable zu erhöhen, können wir folgendermaßen vorgehen:

```
 9    class Program
10    {
11        static void Main(string[] args)
12        {
13            int zahl = 0;
14            zahl += 1;
15
16            Console.WriteLine(zahl);
17            Console.ReadKey();
18        }
19    }
```

Abbildung 6, Eine Variable um 1 erhöhen

Der Code aus Abbildung 6 kann als Abkürzung für die folgende Zeile herangezogen werden:

zahl = zahl + 1;

Diese Methode funktioniert genauso bei den anderen Rechenoperatoren:

zahl -= 1;
zahl *= 1;
zahl /= 1;

Eine weitere Möglichkeit, den Wert einer Variable um 1 zu erhöhen, sehen wir in der nächsten Abbildung:

```
 9    class Program
10    {
11        static void Main(string[] args)
12        {
13            int zahl = 10;
14            zahl++;
15
16            Console.WriteLine(zahl);
17            Console.ReadKey();
18        }
19    }
```

Abbildung 7, Eine Variable um 1 erhöhen

DER MODULO-OPERATOR

Den Modulo-Operator kann man auch als Restwert-Operator bezeichnen, da er uns den Rest einer Division ausrechnet. Um ihn zu verwenden, schreiben wir das **%-Zeichen**.

```
 9    class Program
10    {
11        static void Main(string[] args)
12        {
13            int zahl = 6 % 4; // 1 REST: 2
14            int zahl2 = 10 % 3; // 3 REST: 1
15
16            Console.WriteLine(zahl);
17            Console.WriteLine(zahl2);
18            Console.ReadKey();
19        }
20    }
```

Abbildung 8, Der Modulo-Operator im Beispiel

Output:

Abbildung 9, Der Restwert wird ausgegeben