

Grundlagen der Programmierung in C# – Zählschleifen

Zählschleifen werden in der Programmierung benutzt, wenn man Anweisungen wiederholt ausführen möchte und klar ist, wie oft das sein soll:

```

1 for ( Initialisierung; Bedingung; Iteration)
2 {
3     // Schleifenkörper
4 }
```

Eine Zählschleife besteht dabei aus einer FOR-Anweisung und dem *Schleifenkörper*. Die FOR-Anweisung wird zur Steuerung der Schleife genutzt. Im Schleifenkörper stehen die in jedem Schleifendurchlauf auszuführenden Anweisungen. Dieser wird, wie üblich, mit geschweiften Klammern gekapselt. Besteht der Schleifenkörper aus nur einer einzigen Anweisung, können die geschweiften Klammern entfallen.

Bsp.:

```

1 for (int i=1; i<=5; i++)
2     Console.WriteLine(i);
```

Eine Zählschleife ist am Schlüsselwort FOR zu erkennen. Danach folgt eine Klammer, die aus drei Teilen besteht:

- Initialisierung: Um die benötigte Anzahl der Schleifendurchläufe zu realisieren, benötigt man eine sogenannte *Schleifenvariable*, welche die Durchläufe zählt. Deklaration und Initialisierung so einer Variablen können an dieser Stelle erfolgen.
- Bedingung: Der zweite Abschnitt ist der Abbruchbedingung der Schleife vorbehalten. Hier wird definiert, wie lange die Schleife durchlaufen werden soll. Für gewöhnlich steht an dieser Stelle ein Vergleich der Schleifenvariable.

Vor jedem Schleifendurchlauf wird diese Bedingung überprüft. Die Anweisungen der Schleife werden nur dann ausgeführt, wenn die Bedingung *wahr* ist, im Beispiel also wenn *i* nicht größer als 5 ist.

- Iteration: An dieser Stelle kann festgelegt werden, was **nach einem Schleifendurchlauf** mit der Schleifenvariablen passiert. In obigen Beispiel wird die Schleifenvariable um 1 erhöht.

Achtung C# nutzt für Standardrechnungen Verkürzungen, z. B. ist `i++`; gleichzusetzen mit `i = i + 1`; bzw. `i += 1`; Analog gibt es auch die Kurzform `i--`;

Über die FOR-Anweisung kann die gesamte Schleifensteuerung getätigt werden: Man legt die Schleifenvariable in der Initialisierung fest, erhöht sie sukzessive um 1 nach jedem Schleifendurchlauf und bricht ab (mithilfe der Bedingung), wenn die Schleife oft genug wiederholt bzw. wenn die Schleifenvariable zu groß geworden ist. Im obigen Beispiel wird die Schleife genau fünfmal durchgeführt. Da im Schleifenkörper der aktuelle Wert der Schleifenvariable ausgegeben wird, sieht man, zum wievielten Mal die Schleife durchlaufen wurde.

Achtung Die FOR-Anweisung kann auch leer bleiben: `for (; ;)`. Dann muss allerdings andersweitig sichergestellt werden, dass es zu keiner Endlosschleife kommt, d. h. der Schleifendurchlauf an irgendeiner Stelle abgebrochen wird!

Eine Schleife kann jederzeit mit der Anweisung BREAK; komplett beendet werden.

Beispiele:

```

1 // wiederholte Ausgabe eines Satzes
2 Console.WriteLine("Wie oft soll der Satz ausgegeben werden?");
3 string eingabe = Console.ReadLine();
4
5 int anzahl = Convert.ToInt32(eingabe);
6
7 for (int i = 1; i <= anzahl; i++)
8     Console.WriteLine("Programmieren ist ganz einfach.");

```

Variante I:

```

1 // wiederholte Ausgabe eines Satzes
2 Console.WriteLine("Wie oft soll der Satz ausgegeben werden?");
3 string eingabe = Console.ReadLine();
4
5 int anzahl = Convert.ToInt32(eingabe);
6 int i = 1;
7 for ( ; i <= anzahl; i++)
8     Console.WriteLine("Programmieren ist ganz einfach.");

```

Variante II:

```

1 // wiederholte Ausgabe eines Satzes
2 Console.WriteLine("Wie oft soll der Satz ausgegeben werden?");
3 string eingabe = Console.ReadLine();
4
5 int anzahl = Convert.ToInt32(eingabe);
6 int i = 1;
7
8 for ( ; i <= anzahl; )
9 {
10     Console.WriteLine("Programmieren ist ganz einfach.");
11     i++;
12 }

```

Alternative III:

```

1 // wiederholte Ausgabe eines Satzes
2 Console.WriteLine("Wie oft soll der Satz ausgegeben werden?");
3 string eingabe = Console.ReadLine();
4
5 int anzahl = Convert.ToInt32(eingabe);
6 int i = 1;
7
8 for ( ; ; )
9 {
10     Console.WriteLine("Programmieren ist ganz einfach.");
11     i++;
12     if (i > anzahl)
13         break;
14 }

```

```

1 // Ausgabe aller geraden Zahlen bis 10
2 for (int i = 2; i < 11; i += 2)
3     Console.WriteLine(i);

```

Aufgabe 1. Schreibe ein Programm, welches die Zahlen 1 bis n aufaddiert und diese Summe ausgibt.

Aufgabe 2. Erstelle ein Programm zur Berechnung der Potenz a^b , nachdem der Nutzer den natürlichen Exponent b und die reelle Basis a eingegeben hat.

Aufgabe 3. Schreibe ein Programm, das die Zahlwörter eins, zwei, ..., zehn so oft ausgibt, wie das Wort aussagt, also eins einmal, zwei zweimal, ...