

Die Mehrfachverzweigung mit else if..

In C# ermöglicht die Mehrfachverzweigung mit `else if` die Untersuchung mehrerer Bedingungen nacheinander und die Ausführung von Anweisungen, basierend darauf, welche Bedingung zuerst als `true` evaluiert wird. Es wird oft verwendet, um komplexe Entscheidungsstrukturen in Programmen zu implementieren.

Hier ist die grundlegende Syntax der Mehrfachverzweigung mit `else if`:

```
1  if (Bedingung1)
2  {
3  |    // Anweisungen, die ausgeführt werden, wenn Bedingung1 wahr ist
4  | }
5  else if (Bedingung2)
6  {
7  |    // Anweisungen, die ausgeführt werden, wenn Bedingung2 wahr ist
8  | }
9  else if (Bedingung3)
10 {
11 |    // Anweisungen, die ausgeführt werden, wenn Bedingung3 wahr ist
12 | }
13 ...
14 else
15 {
16 |    // Anweisungen, die ausgeführt werden, wenn keine der obigen Bedingungen wahr ist
17 | }
```

Ein konkretes Beispiel:

```
1  int zahl = 25;
2
3  if (zahl < 10)
4  {
5  |    Console.WriteLine("Die Zahl ist kleiner als 10.");
6  | }
7  else if (zahl < 20)
8  {
9  |    Console.WriteLine("Die Zahl ist kleiner als 20 aber größer oder gleich 10.");
10 | }
11 else if (zahl < 30)
12 {
13 |    Console.WriteLine("Die Zahl ist kleiner als 30 aber größer oder gleich 20.");
14 | }
15 else
16 {
17 |    Console.WriteLine("Die Zahl ist 30 oder größer.");
18 | }
```

In diesem Beispiel wird die Ausgabe "Die Zahl ist kleiner als 30 aber größer oder gleich 20." sein.

Einige wichtige Punkte:

1. Die `if`-Bedingungen werden in der Reihenfolge überprüft, in der sie erscheinen. Sobald eine Bedingung als `true` evaluiert wird, werden die Anweisungen in diesem Block ausgeführt und die restlichen `else if`- oder `else`-Blöcke übersprungen.
2. Der `else`-Block am Ende ist optional. Er wird nur dann ausgeführt, wenn keine der vorherigen `if`- oder `else if`-Bedingungen als `true` evaluiert wurde.
3. Es gibt keine Begrenzung für die Anzahl der `else if`-Blöcke, die Sie verwenden können.