

C# Einfache Anweisungen

Leeranweisung

```
; // ; ist Terminator, nicht Separator
```

Wertzuweisung

```
x = 3 * y + 1;
```

Methodenaufruf

```
string s = "a,b,c";  
string[] parts = s.Split(','); // Aufruf einer Objektmethode (nicht static)  
  
s = Console.ReadLine(); // Aufruf einer Klassenmethode (static)
```

C# if-Anweisung

```
if ('0' <= ch && ch <= '9')
    val = ch - '0';
else if ('A' <= ch && ch <= 'Zx)
    val = ch - 'A';
else
{
    val = 0;
    Console.WriteLine("invalid character " + ch);
}
```

C# switch-Anweisung

```
switch (country)
{
    case "Germany": case "Austria": case "Switzerland":
        language = "German";
        break;
    case "England": case "USA":
        language = "English";
        break;
    case null:
        Console.WriteLine("kein Land ausgewaehlt");
        break ;
    default:
        Console.WriteLine("kenne die Sprache des Landes {0} nicht", country);
        break ;
}
```

Switch-Ausdruck muß **numerisch**, **char**, **enum** oder **string** sein (null erlaubt).

Case-Ausdrücke müssen mit Switch-Ausdruck kompatibel sein.

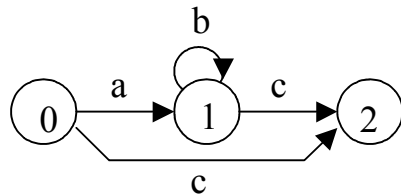
Switch mit string-Marken wird sequentiell geprüft.

Jede Case-Anweisungsfolge muß mit **break** (oder **return**, **goto**, **throw**) enden.

Wenn kein Ausdruck passt, wird zu default verzweigt, default-Zweig ist optional

C# switch mit Sprüngen

Z.B. zur Implementierung endlicher Automaten



```

int state = 0;
int ch = Console.Read();
switch (state) {
    case 0: if (ch == 'a') { ch = Console.Read(); goto case 1; }
           else if (ch == 'c') goto case 2;
           else goto default;
    case 1: if (ch == 'b') { ch = Console.Read(); goto case 1; }
           else if (ch == 'c') goto case 2;
           else goto default;
    case 2: Console.WriteLine("input valid");
           break;
    default: Console.WriteLine("illegal character " + ch);
            break;
}
  
```

C# Schleifen

while

```
while (i < n)
{
    sum += i;
    i++;
}
```

do while

```
do
{
    sum += a[i];
    i--;
} while(i > 0);
```

for

```
for (int i = 0; i < n; i++)
    sum += i;
```

C# foreach-Anweisung

Zum Abarbeiten beliebiger Collections und Arrays

```
int[] a = {3, 17, 4, 8, 3, 29};  
foreach (int x in a) sum += x;
```

```
string s = "Hello";  
foreach (char ch in s) Console.WriteLine(ch);
```

```
Queue q = new Queue();  
q.Enqueue("John"); q.Enqueue("Alice"); ...  
foreach (string s in q) Console.WriteLine(s);
```

C# Sprünge

- break;** Zum Ausprung aus Schleifen und switch-Anweisungen.
- continue;** Wird in Schleifen verwendet, um an den *Schleifenanfang* zurückzuspringen.
- goto case 3:** Kann in einer switch-Anweisung zum Ansprung einer der case-Marken verwendet werden;
- myLab:
...
- goto myLab;** Springt zur Marke myLab.
Restriktion:
- kein Einsprung in einen Block

C# return-Anweisung

Aussprung aus Methoden

```
void fkt (int x)
{   if (x == 0) return;
    ...
}
```

Aussprung aus Funktionsmethoden

```
int max(int a, int b) {
    if (a > b) return a; else return b;
}

class C
{   static int Main()
    {.....
    return errorCode;    // Rückgabe eines Fehlercodes in der Main-Methode
    }                   // eines Programms (kann in Dos-Box mittels Variable
                        // errorlevel abgeprüft werden).
}
```