```
// Syntax der switch-Anweisung

switch(Ausdruck) {
  case Wert1 :
    // Anweisungen
    break;
  case Wert 2 :
    // Anweisungen
    break;
  [default:
    // Anweisungen
    break;]
}
```

Ausdruck		
Wert1	Wert2	default (sonst)
Anweisungen	Anweisungen	Anweisungen
break	break	break

```
private void Wiederhole(int anzahl)
{
    for(int i=0; i< anzahl; i++)
    {
        Console.WriteLine(i);
    }
}
```

```
String[] wochentage = {"Mo","Di","Mi","Do","Fr","Sa","So");
foreach(string tag in wochentage)
{
       Console.WriteLine($"{tag");
}
```

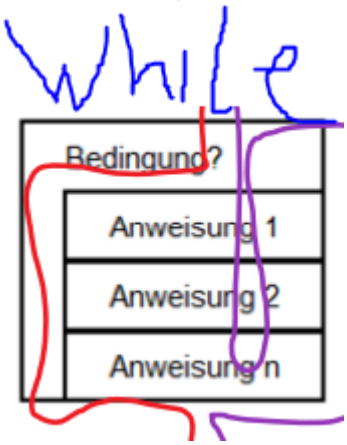
Ausgabe:

Mo
Di
Mi
Do
Fr
Sa
So

Die Schleife läuft 7-mal durch: der Array wochentage hat 7 Elemente. Mit der lokalen Variablen tag wird auf jedes der Elemente zugegriffen.

Methode	Beschreibung
`ToBoolean` (Ausdruck)	Konvertiert den Ausdruck in einen `bool`-Typ.
`ToByte` (Ausdruck)	Konvertiert den Ausdruck in einen `byte`-Typ.
`ToChar` (Ausdruck)	Konvertiert den Ausdruck in einen `char`-Typ.
`ToDecimal` (Ausdruck)	Konvertiert den Ausdruck in einen `decimal`-Typ.
`ToDouble` (Ausdruck)	Konvertiert den Ausdruck in einen `double`-Typ.
`ToInt16` (Ausdruck)	Konvertiert den Ausdruck in einen `short`-Typ.
`ToInt32` (Ausdruck)	Konvertiert den Ausdruck in einen `int`-Typ.
`ToInt64` (Ausdruck)	Konvertiert den Ausdruck in einen `long`-Typ.
`ToSByte` (Ausdruck)	Konvertiert den Ausdruck in einen `sbyte`-Typ.
`ToSingle` (Ausdruck)	Konvertiert den Ausdruck in einen `float`-Typ.
`ToString` (Ausdruck)	Konvertiert den Ausdruck in einen `string`-Typ.
`ToUInt16` (Ausdruck)	Konvertiert den Ausdruck in einen `ushort`-Typ.
`ToUInt32` (Ausdruck)	Konvertiert den Ausdruck in einen `uint`-Typ.
`ToUInt64` (Ausdruck)	Konvertiert den Ausdruck in einen `ulong`-Typ.

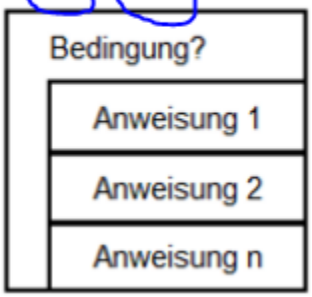
```
do
{
    Anweisung 1;
    Anweisung 2;
    Anweisung n;
} while(Bedingung);
```

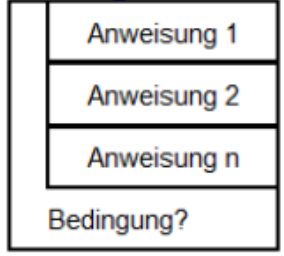



```
i++;
i = i + 1;
```

```
i += 5;
i = i + 5;
```

```
i *= 2;
i = i * 2;
```





Variablen

Eine Variable ist eine Möglichkeit, um im Programm eine Zahl, String (Buchstaben- und Zahlenwert mit dem nicht gerechnet werden kann) oder einen boolschen Wert (wahr oder falsch) zu speichern. In fast allen Programmiersprachen sind Variable der wichtigste Teil einer Programmiersprache. Oft ist es nur durch diese "Platzhalter" möglich, die durch einen eingegebenen Wert ersetzt werden können, interaktive Programme zu erstellen.

Verzweigung:

Verzweigung

Eine Verzweigung ist eine Stelle in einem Programm, wo es unterschiedliche Möglichkeiten gibt, mit dem Programm fortzufahren, je nachdem ob eine gestellte Bedingung zutrifft oder nicht.

Schleifen:

Schleife

Eine Schleife enthält Code der solange ausgeführt wird, wie die Bedingung, die zur Schleife gehört wahr ist.

Prozedur oder Funktion (auch: Methode):

Methode

Eine Methode ist eine Sammlung von Code, die jederzeit aufgerufen werden kann. Z.B. für öfters anfallende Berechnungen schreibt man den Code in eine Funktion, so muss der Code nur einmal geschrieben werden und kann einfach über den jeweiligen Funktionsnamen aufgerufen werden.

Methodendeklaration

```
//Hier können Sie eigenen Methoden einfügen

private void AvoidTree()
{
    kara.TurnLeft();
    kara.Move();
    kara.TurnRight();
    kara.Move();
    kara.Move();
    kara.TurnRight();
    kara.Move();
    kara.TurnLeft();
}//Ende Methode GeheUmBaumHerum
}
```

Methodendeklaration mit
Methodenkopf
und
Methodenrumpf

```
private void Main(string args)
{
    //Hier die Anweisungen notieren

    for (int i = 0; i < 5; i++)
    {
        kara.PutLeaf();
        kara.Move();
    }

    //Programmende: nächste Zeile bitte nicht löschen
    MessageBox.Show("Programmcode ausgeführt!");
```

```
//Hier können Sie eigenen Methoden einfügen
private int CountLeafs(int laenge)
{
    int anzahl = 0;
    for (int i = 0; i < laenge; i++)
    {
        if (kara.OnLeaf())
        {
            anzahl++;
        }
        kara.Move();
    }
    return anzahl; // Rueckgabe des Funktionswertes
} // Ende von CountLeafs
```

Der Rückgabetyp muss gleich dem Typ des Ausdrucks nach "return" sein!

Sie sollen die Zahlen 3.1, 27.88, 45.45 und 66.6 abspeichern.

```
double[] zahlen ={3.1, 27.88, 45.45, 66.6};
```

Die Buchstaben des Wortes "Programmieren" sollen einzeln abgespeichert werden.

```
char[] buchstaben ={'P', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'e', 'r',
                    'e', 'n' };
```

Weisen Sie dem 10. Element des Arrays buchtitel den Wert "Zahlensysteme" zu.

```
buchtitel[9]="Zahlensysteme";
```

Weisen Sie das 1. Element des Arrays kommazahlen einer neu zu deklarierenden Variablen zu.

```
double zahl = kommazahlen[0];
```

Vergleichen Sie, ob das 1. Element des Arrays ergebnis grösser ist als das letzte. Wenn ja, geben Sie den Text "Das erste ist grösser!" auf der Konsole aus.

```
If(ergebnis[0] > ergebnis[ergebnis.Length -1])
{
    Console.WriteLine("Das erste ist grösser!")
}
```

<	kleiner als	>=	grösser oder gleich
>	grösser als	==	gleich
<=	kleiner oder gleich	!=	ungleich

&	UND mit vollständiger Auswertung: beide Teilbedingungen werden auf jeden Fall ausgewertet
&&	UND mit kurzer Auswertung: ist die erste Teilbedingung bereits falsch, wird die zweite Teilbedingung nicht ausgewertet, da das Gesamtergebnis bereits feststeht
|	ODER mit vollständiger Auswertung: beide Teilbedingungen werden auf jeden Fall ausgewertet
||	ODER mit kurzer Auswertung: ist die erste Teilbedingung bereits richtig, wird die zweite Teilbedingung nicht ausgewertet, da das Gesamtergebnis bereits feststeht
^	exklusives ODER
!	NICHT

.NET-Laufzeittyp	C#-Alias	CLS-konform	Wertebereich
Byte	byte	ja	0 ... 255
SByte	sbyte	nein	-128 ... 127
Int16	short	ja	-2^{15} ... 2^{15} -1
UInt16	ushort	nein	0 ... 65535
Int32t	int	ja	-2^{31} ... 2^{31} -1
UInt32	uint	nein	0 ... 2^{32} -1
Int64	long	ja	-2^{63} ... 2^{63} -1
UInt64	ulong	nein	0 ... 2^{64} -1
Single	float	ja	$1,4 * 10^{-45}$ bis $3,4 * 10^{38}$
Double	double	ja	$5,0 * 10^{-324}$ bis $1,7 * 10^{308}$
Decimal	decimal	ja	+/-79E27 ohne Dezimalpunktangabe; +/-7.9E-29, falls 28 Stellen hinter dem Dezimalpunk Zahl beträgt +/-1.0E-29.
Char	char	ja	Unicode-Zeichen zwischen 0 und 65535
String	string	ja	ca. 2^{31} Unicode-Zeichen
Boolean	bool	ja	true oder false
Object	object	ja	Eine Variable vom Typ Object kann jeden anderen D