

Ein etwas frecher „GrownUpCalculator“

```
PS P:\c#\GrownUpCaculator> .\GrownUpCalculator.exe
How old are you? 15
In approx. 3 years you will be grown-up
PS P:\c#\GrownUpCaculator> .\GrownUpCalculator.exe
How old are you? 22
You are too old to use that program
PS P:\c#\GrownUpCaculator> _
```

Das Programm prüft, ob das eingegebene Alter 18 Jahre beträgt.
Nur dann wird die Differenz ausgerechnet und ausgegeben.

Der einfache Teil

```
using System;
```

```
const int GROWN_UP_AGE = 18;
```

```
Console.Write("How old are you? ");
```

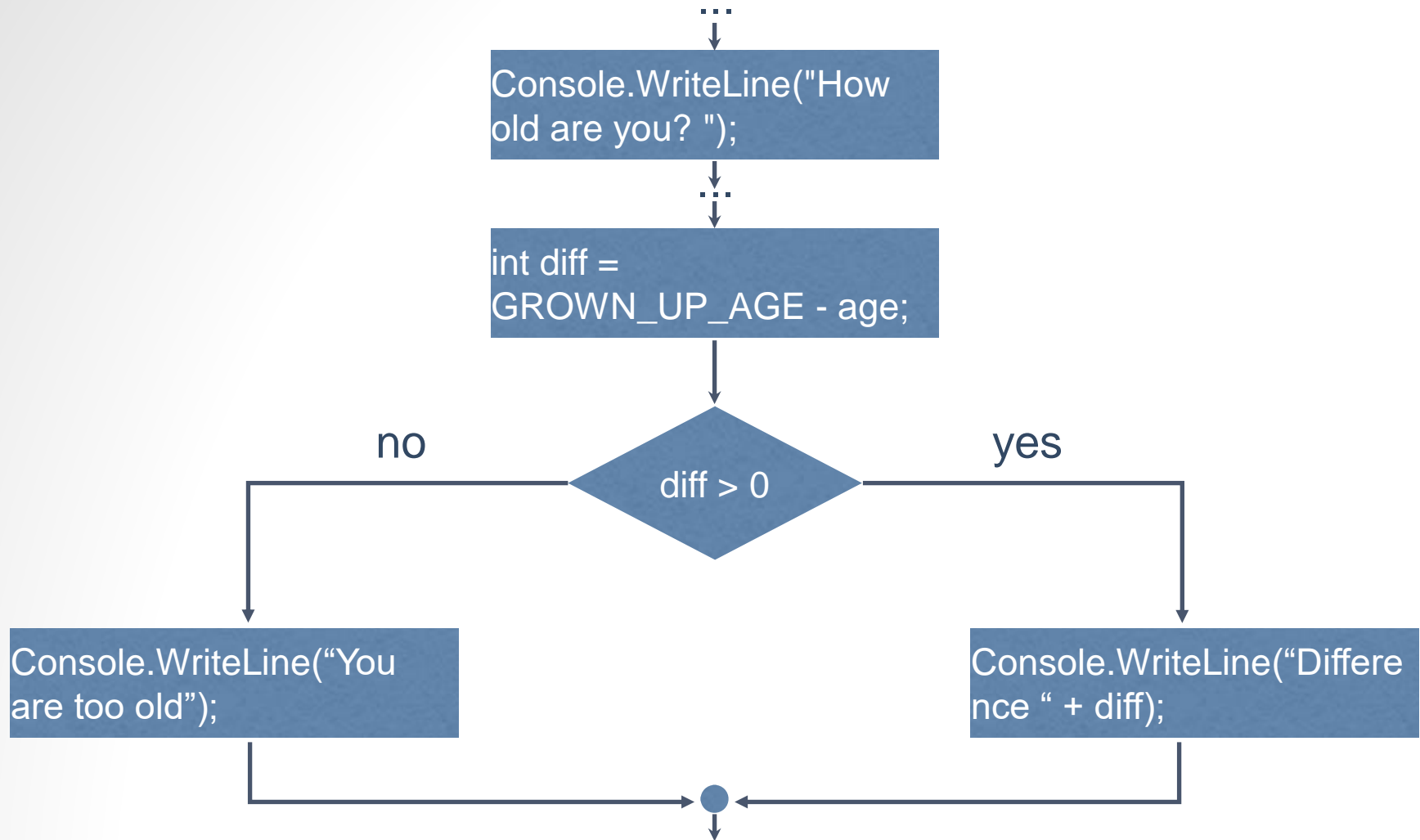
```
string userInput = Console.ReadLine();
```

```
int age = Convert.ToInt32(userInput);
```

```
int diff = GROWN_UP_AGE - age;
```

```
Console.WriteLine($"In approx. {diff} years you will be grown-up");
```

Verzweigung - Selektion



Selektion in C#

```
if (diff > 0)
{
    Console.WriteLine($"In approx. {diff} years you will be grown-up");
}
else
{
    Console.WriteLine("You are too old to use that program");
}
```

Bedingungen (Conditions)

<	Kleiner
>	Größer
<=	Kleiner oder gleich
>=	Größer oder gleich
==	gleich
!=	ungleich

Bedingungen – Logischer Ausdruck

Eine **Bedingung** ist ein logischer Ausdruck.

Ein logischer Ausdruck besteht aus **Operanden**, die mit *relationalen* oder *logischen Operatoren* verglichen werden.

Vergleichsoperatoren (relationale Operatoren):

a == b

a ist gleich b

a != b

a ist nicht gleich b

a > b

a ist größer als b

a >= b

a ist größer oder gleich als b

a < b

a ist kleiner als b

a <= b

a ist kleiner oder gleich als b

Verknüpfung von Bedingungen mit && (AND)

- Situation
 - Wenn das Alter zwischen 18 und 20 Jahren liegt, schreiben wir “You are grown up”
- Siehe Mathematik
 - $\text{if } (18 \leq \text{age} \leq 20)$
 - Das ist die Kurzform für:
 - $\text{if } (18 \leq \text{age} \text{ and } \text{age} \leq 20)$
 - $\text{if } (\text{age} \geq 18 \text{ and } \text{age} \leq 20)$
- **and** in C#

```
if (18 <= age && age <= 20)
{
    Console.WriteLine(...)
}
```

```
if (age >= 18 && age <= 20)
{
    Console.WriteLine(...)
}
```

Neu in c# 9.0

```
if (age is (>= 18 and <= 20))  
{  
    Console.WriteLine("You are grown up");  
}
```


Verknüpfung von Bedingungen mit II (OR)

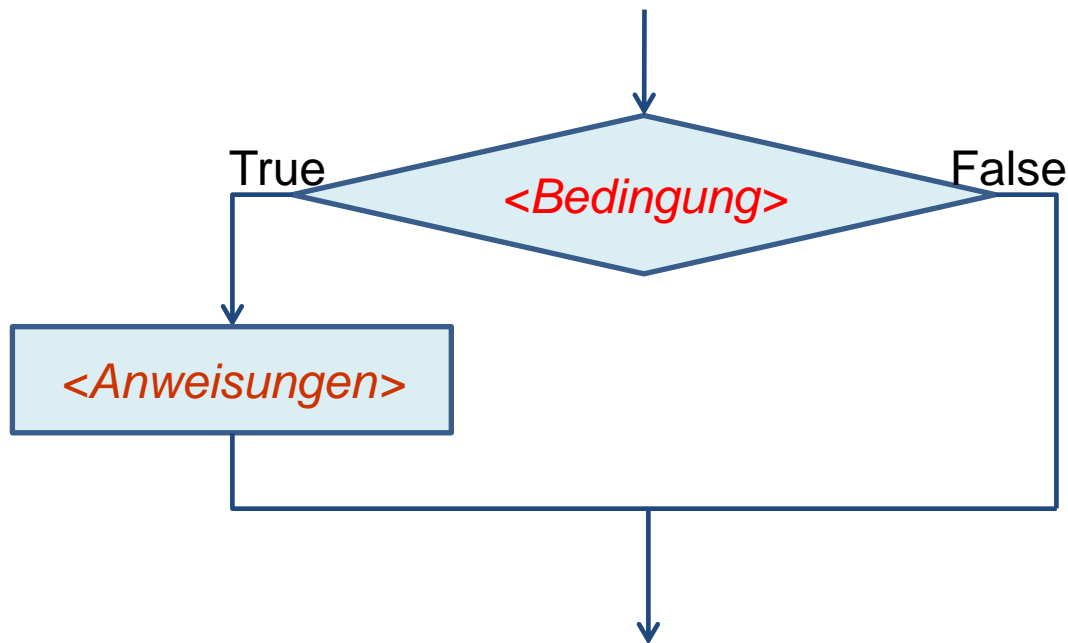
- Situation
 - Wenn das Alter entweder 25 oder 50 ist, schreiben wir “Congratulations for your round birthday”
- Wieder Mathematik
 - if (age == 25 **or** age == 50)
- **or** in C#

```
if (age == 25 || age == 50)
{
    Console.WriteLine("Congratulations for your round birthday");
}
```

Einseitige Verzweigung

Einseitige Verzweigung

Trifft eine *<Bedingung>* zu, so werden bestimmte *<Anweisungen>* ausgeführt.



Einseitige Verzweigung (if)

Einseitige Verzweigung

Trifft eine *<Bedingung>* zu, so werden bestimmte *<Anweisungen>* ausgeführt.

```
if ( <Bedingung> )  
{  
    <Anweisungen>  
}
```

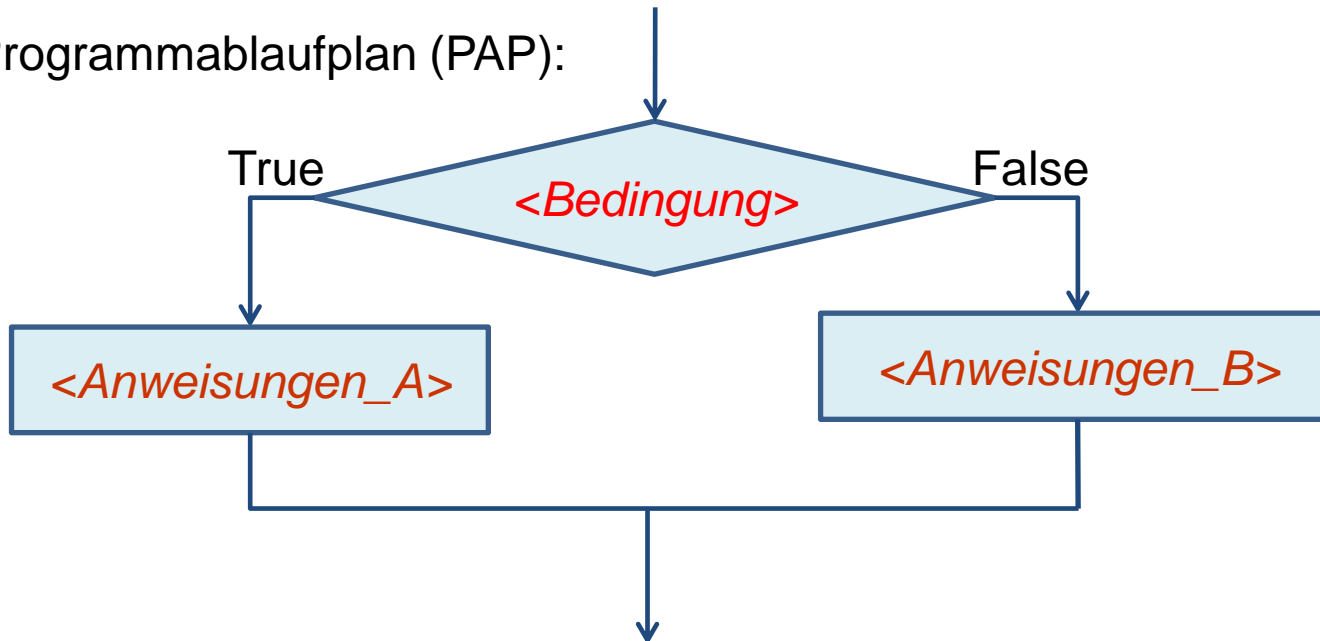
```
string input = Console.ReadLine();  
int mark = Convert.ToInt32(input);  
if (mark > 5)  
{  
    Console.WriteLine($"Die Note {mark} ist nicht möglich.");  
}
```

Zweiseitige Verzweigung (if .. else)

Zweiseitige Verzweigung

Trifft eine *Bedingung* zu, so werden *Anweisungen_A* ausgeführt, trifft sie nicht zu, so werden *Anweisungen_B* ausgeführt.

Programmablaufplan (PAP):



Zweiseitige Verzweigung (if..else)

Zweiseitige Verzweigung

Trifft eine *Bedingung* zu, so werden *Anweisungen_A* ausgeführt, trifft sie nicht zu, so werden *Anweisungen_B* ausgeführt.

```
if ( <Bedingung> )  
{  
    <Anweisungen_A>  
}  
else  
{  
    <Anweisungen_B>  
}
```

Zweiseitige Verzweigung - Beispiel

```
string choice = Console.ReadLine();  
if (choice == "A")  
{  
    Console.WriteLine("Es wurde A gewählt!");  
}  
else  
{  
    Console.WriteLine("Es wurde etwas anderes als A gewählt!");  
}
```

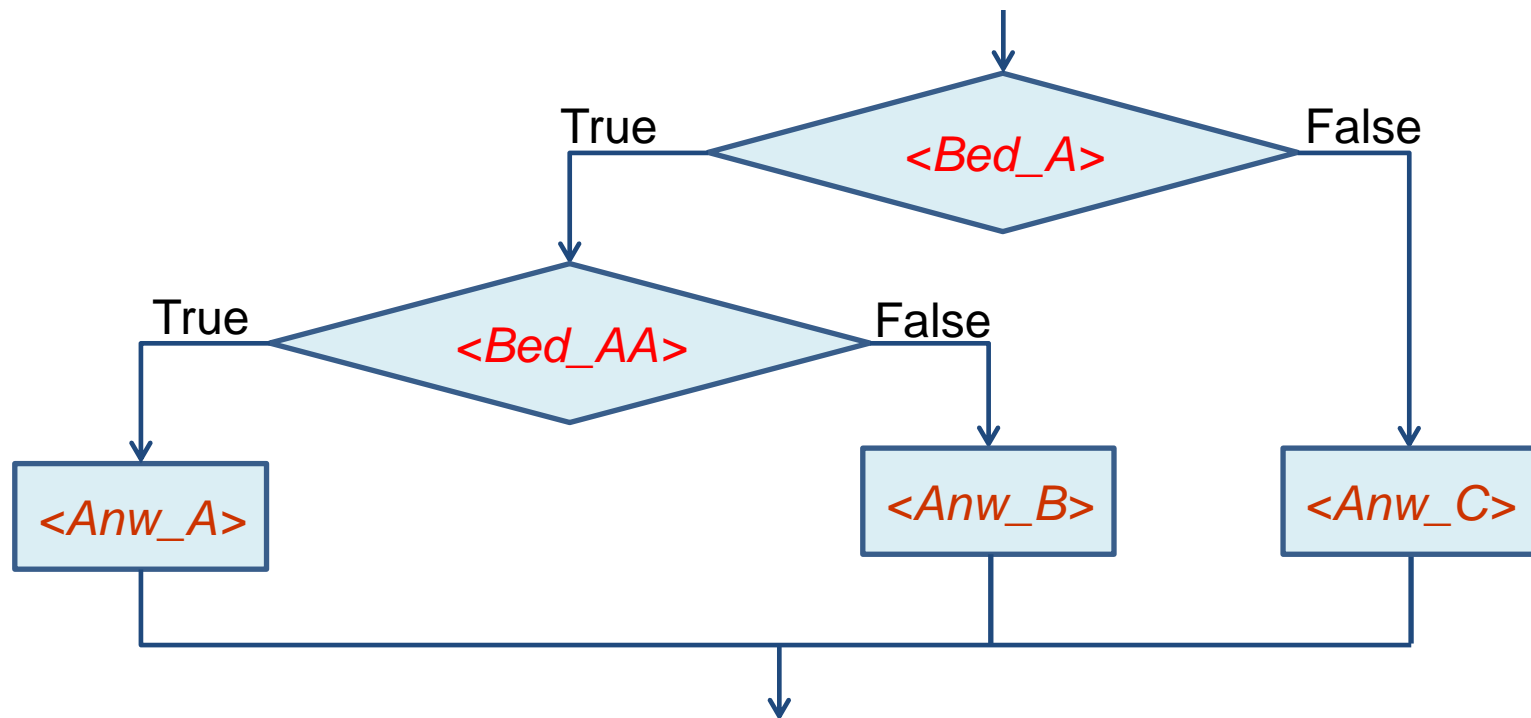
Erweiterung

- Wenn das Alter unter 18 ist
 - "In approx <diff> years you will be grown-up"
- Wenn das Alter zwischen 18 und 20 ist
 - "You are grown up"
- Wenn der User noch älter ist
 - "You are too old"

Verschachtelung von Verzweigungen

Einschränkung

Eine Bedingung **<Bed_A>** wird noch einmal genauer durch eine Bedingung **<Bed_AA>** geprüft.



Verschachtelung von Verzweigungen

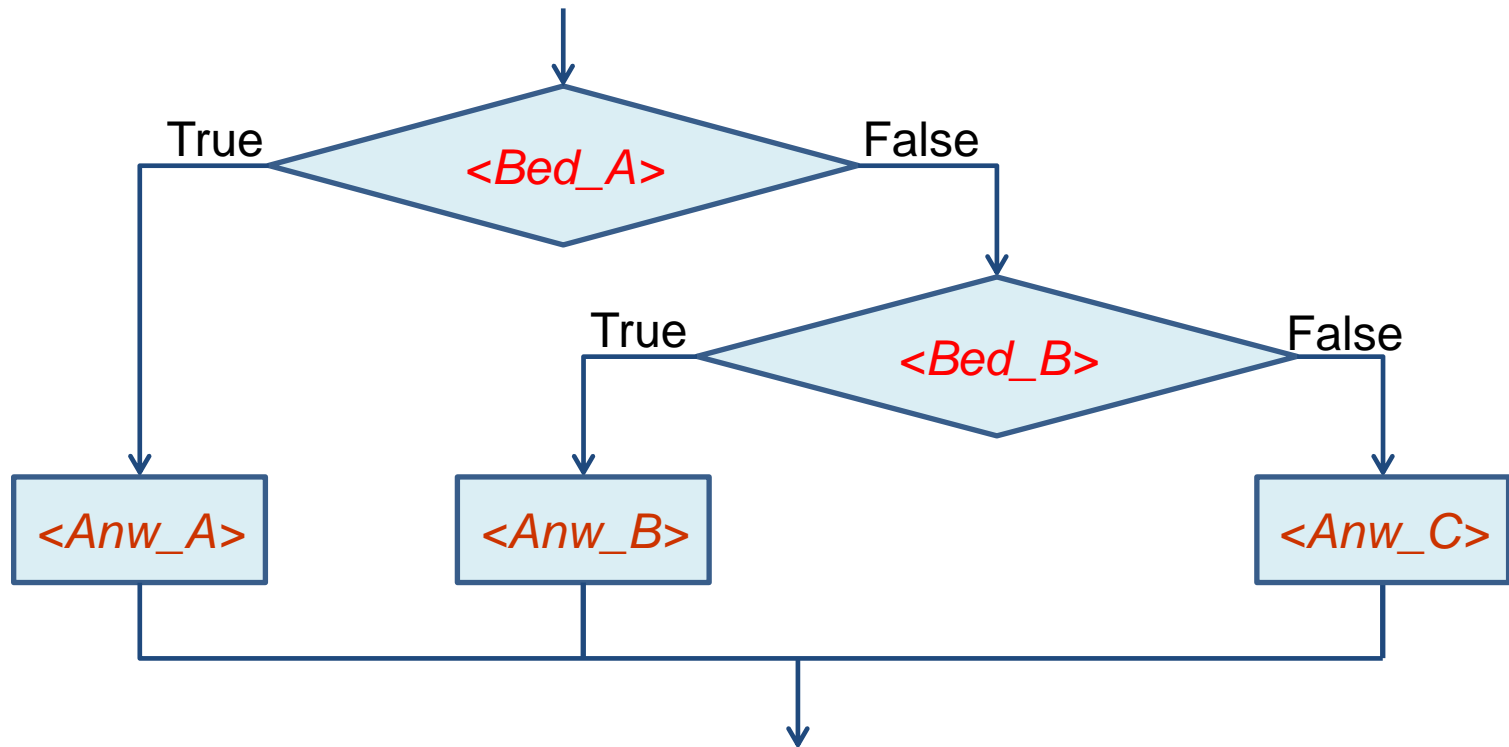
Beispiel

```
int choice = Console.ReadLine();
if (choice == "H")
{
    Console.WriteLine("Hauptschule! Alter ?");
    string input = Console.ReadLine();
    int age = Convert.ToInt32(input);
    if (age == 7)
    {
        Console.WriteLine("1.Klasse Hauptschule");
        ...
    }
    ...
}
```

Verschachtelung von Verzweigungen

Alternative (*else-if* Kette)

Es gibt mehr als zwei Alternativen zur Programmausführung .



Beispiel else-if Kette

Alternative (else-if Kette)

```
string choice = Console.ReadLine();
if (choice == "A")
{
    Console.WriteLine("AHS");
}
else if (choice == "B")
{
    Console.WriteLine("Hauptschule");
}
else if (choice == "C")
{
    Console.WriteLine("Neue Mittelschule");
}
else
{
    Console.WriteLine("ungültige Eingabe");
}
```

Noch mehr Varianten

- Neue Anforderungen

- Wenn das Alter 15 ist schreiben wir “Hey, you could be my brother”
- Wenn das Alter 40 ist schreiben wir “Hey, you could be my father”
- Wenn das Alter 65 ist schreiben wir “Hey, you could be my grand father”
- Wenn das Alter 80 ist schreiben wir “HEY, GRAND GRAND FATHER!!”
- Andernfalls schreiben wir “No silly comment available for your age”

or sister ...

Mother?


Lauter sprechen, hört nicht mehr so gut ...

- Lösung mit if?

- Naja, es ist möglich
- Aber besser mit Switch!

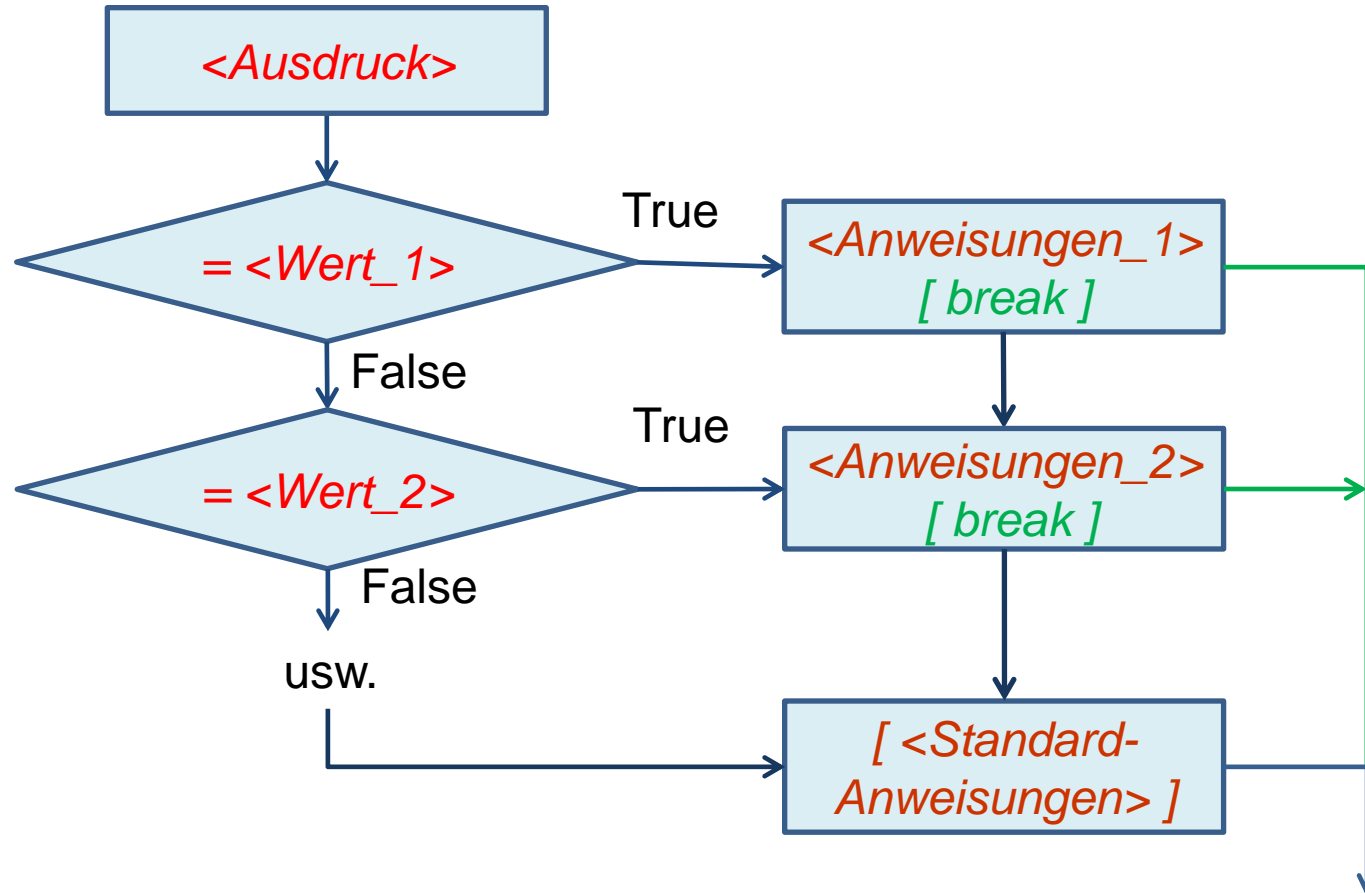
Switch in C#

```
switch (age)
{
    case 15:
        Console.WriteLine("Hey you could be my brother");
        break;
    case 40:
        Console.WriteLine("Hey you could be my father");
        break;
    ...
    default:
        Console.WriteLine("No silly comment available");
        break;
}
```



Gleiches Muster für die anderen Fälle

Mehrfachverzweigung (switch)



Mehrfachverzweigung (switch)

Ergibt die Auswertung von *Ausdruck* den Wert *Wert_N*, so werden *Anweisungen_N* ausgeführt. Wird keine Übereinstimmung gefunden, so werden *<Standard_Anweisungen>* ausgeführt, wenn das Schlüsselwort **default** angegeben wird.

Ein **break** muss nach jedem case-Fall und am Ende des default-Falles angeführt werden!

```
switch ( <Ausdruck> )
{
    case <Wert_1> : <Anweisungen_1>
                    break;
    case <Wert_2> : <Anweisungen_2>
                    break;
    . . .
    [ default : <Standard_Anweisungen>
                break; ]
}
```