

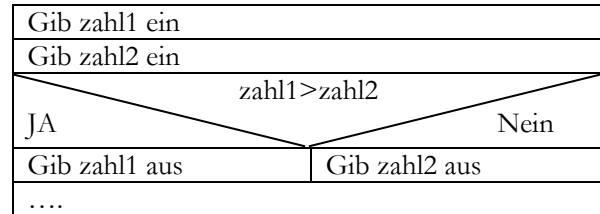
2.3 Verzweigungen

Mit den bisherigen Befehlen ist es nur möglich linear zu programmieren, d.h. ein Befehl folgt auf den anderen. Die meisten Probleme erfordern aber leider etwas andere Lösungen. In Abhängigkeit von einer Bedingung wird entweder der eine Programmteil, oder der andere Programmteil ausgeführt.

Bsp. 10: Die größere von zwei Zahlen ausgeben.

Die nebenstehende Abbildung wird als Struktogramm (Nassi-Shneiderman-Diagramm) bezeichnet. Man kann hier den Programmablauf erkennen.

Wenn die Bedingung erfüllt ist, wird der Programmteil JA ausgeführt sonst der Teil NEIN.



```
static void Main(string[] args)
{int a,b;
  Console.Write("a? ");a=Convert.ToInt32(Console.ReadLine());
  Console.Write("b? ");b=Convert.ToInt32(Console.ReadLine());

  if (a>b)
  {
    Console.WriteLine("Größere Zahl:{0}",a);
  }
  else
  {
    Console.WriteLine("Größere Zahl:{0}",b);
  }
  Console.ReadLine();
}
```

Die Syntax für die if-Verzweigung schaut im einfachsten Fall so aus:

```
if (Bedingung)
{
  Anweisungsblock für JA
}
else
{
  Anweisungsblock für NEIN
}
```

Operator	Funktion
= =	gleich
<	kleiner
<=	kleiner gleich
>=	größer gleich
>	größer
!=	ungleich

Der *else* – Teil kann auch weggelassen werden. Beachten Sie, dass Sie mit einem == auf Gleichheit überprüfen.

Wie man in der Tabelle sieht, kann in einer Bedingung nicht nur auf Gleichheit überprüft werden.

Ü 11: *Rabattberechnung: Eingabe: Preis pro Stück, Anzahl. Ist die Anzahl über 100 Stück soll ein Rabatt von 10% gewährt werden.*
Ausgabe: Endsumme

Viele Programmierer glauben, ihren Code möglichst unverständlich schreiben zu müssen. Schnellere Laufzeit und weniger benötigter Speicherplatz werden meist als Gründe dafür angeführt. Ich glaube jedoch, dass das Gebot der Programmierung die gute Lesbarkeit und Nachvollziehbarkeit des Codes sein sollte. Man gewinnt für die Laufzeit eines gesamten Softwareprojektes, von der Analyse über die Implementierung bis hin zur Wartung an Änderungsfreundlichkeit und Übersicht. Geschwindigkeit spielt nur in einem kleinen Teil von Programmen eine herausragende Rolle. Deshalb werden Sie hier im Skriptum meist nicht die kürzesten, aber dafür (hoffentlich) verständliche Algorithmen finden. Sie können natürlich selbst nach anderen Lösungen suchen, vergessen Sie dabei aber nicht ihre „Tricks“ zu dokumentieren.

Mehrere Bedingungen

Manchmal ist es notwendig, **mehrere Bedingungen** miteinander zu verknüpfen. Prinzipiell gibt es zwei Möglichkeiten:

- &&** beide Bedingungen müssen wahr sein, damit der Anweisungsbock JA durchlaufen wird.
- ||** mindestens eine der Bedingungen muss wahr sein, damit der Anweisungsbock JA durchlaufen wird

```

if (Bedingung1) || (Bedingung2)
{...
}
else
{...
}

```

```

if (Bedingung1) && (Bedingung2)
{...
}
else
{...
}

```

Bsp. 11: Für eine Umfrage liegt die Zielgruppe zwischen 18 und 33 Jahren. Es soll das Alter eingegeben werden. Als Ausgabe erscheint "in der Zielgruppe" oder "außerhalb der Zielgruppe"

```

static void Main(string[] args)
{int alter;
  Console.WriteLine("Alter? ");alter=Convert.ToInt32(Console.ReadLine());

  if (alter>=18 && alter<=33)
  {
    Console.WriteLine("Mit {0}Jahren ist man in der Zieleguppe",alter);
  }
  else
  {
    Console.WriteLine("Mit {0}Jahren ist man außerhalb der Zielgruppe",alter);
  }
  Console.ReadLine();
}

```

Mehrere Verzweigungsmöglichkeiten

Bsp. 12: Farbige Ausgaben: Sie geben ein Temperatur ein, unter 0°C ist die Ausgabe Blau, über 30° Rot. der Hintergrund ist Weiß.

```

static void Main(string[] args)
{
  double Temperatur;
  Console.BackgroundColor = ConsoleColor.White; Console.Clear();
  Console.ForegroundColor = ConsoleColor.Black;
  Console.WriteLine("Bitte geben Sie eine Temperatur ein? ");
  Temperatur = Convert.ToDouble(Console.ReadLine());
  if (Temperatur < 0)
  {
    Console.ForegroundColor = ConsoleColor.Blue; Console.WriteLine("kalt - unter 0°C");
  }
  else
  {
    if (Temperatur <= 30)
    {
      Console.WriteLine("Normaltemperatur");
    }
    else
    {
      Console.ForegroundColor = ConsoleColor.Red; Console.WriteLine("warm - über 30°C");
    }
  }
  Console.ReadLine();
}

```

Damit die Hintergrundfarbe funktioniert, muss der Bildschirm gelöscht werden (Console.Clear()).