## 2. Boolesche Variable und Ausdrücke[1]

Bei der Ausführung verzweigter Algorithmen muss der Rechner offensichtlich Entscheidungen zwischen verschiedenen Alternativen treffen. Wie kann aber eine Maschine, die nur die Zustände 0 und 1 unterscheiden kann, Entscheidungen zwischen verschiedenen Alternativen treffen? Dazu muss bei einer verzweigten Anweisung noch vor der Ausführung des `if` – bzw. `else` – Zweiges die Bedingung auf ihre Gültigkeit überprüft werden. Dabei wird festgestellt, ob die nach `if` stehende Aussage wahr oder falsch ist. Dazu benötigt man einen Ausdruck, der nur die Werte `false` (falsch) oder `true` (wahr) annehmen kann. Solche Ausdrücke heißen *boolesche Ausdrücke*[2]. Ausdrücke, die Vergleichsoperatoren wie <, > =, etc. enthalten sind ebenfalls Boolesche Ausdrücke.
Oftmals müssen auch Variable verwendet werden, die nur die beiden Werte wahr und falsch annehmen können. Die Vereinbarung von solchen Variablen des Typs `Boolean` erfolgt folgendermaßen:

```
var variable1, variable2, ...: BOOLEAN;
```

Mit Hilfe solcher Boolescher Variablen und Ausdrücke können nun logische Operationen ausgeführt werden.

Aufgabe: Vervollständigen Sie die Tabelle!

| Satz 1 | Satz 2 | NOT Satz1 | Satz1 AND Satz2 | Satz1 OR Satz2 |
|---|---|---|---|---|
| 2 < 5 *true* | 5 ist Primzahl (PZ) *true* | 2 > 5 *false* | 2 < 5 und 5 ist PZ *true* | 2 < 5 oder 5 ist PZ *true* |
| 2 < 5 | 4 ist PZ | | | |
| 7 < 5 | 11 ist PZ | | | |
| 7 < 5 | 12 ist PZ | | | |

Wahrheitswerttabelle:

| **Variable** | | **Operatoren** | | | |
|---|---|---|---|---|---|
| **x** | **y** | **NOT x** | **x AND y** | **x OR y** | **x XOR y** **(Addition modulo 2)** |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |

[1] Inhaltlich entnommen aus: Junger, Rapp, Rathfelder, Reinelt, *Informatik*, Diesterweg / Sauerländer – Verlag, 1988
[2] George Boole (1815 – 1864), englischer Mathematiker

Aufgabe: Bestimmen Sie den Wahrheitswert der folgenden Booleschen Ausdrücke!

- c := NOT (x OR y) mit x := true und y := false
- a := (x OR y) AND NOT (x) mit x := false und y := true
- x := a OR (b AND b) mit a := false und b := true
- y := (a OR b) AND (a OR (NOT(a))) mit a := true und b := false
- x := (a OR b) AND (b OR a) mit a := false und b := false

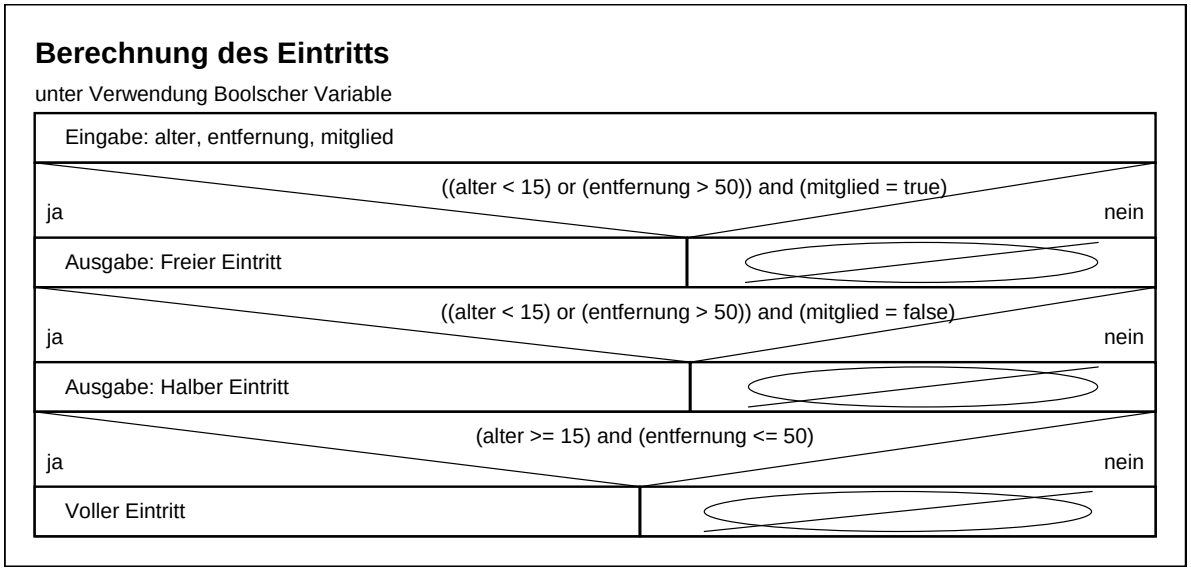
Beispielaufgabe: Realisierung verknüpfter Bedingungen mit Hilfe von Booleschen Variablen

Am Eingang eines Sportplatzes wird ein Informationscomputer aufgestellt. Nach Eingabe des Alters, der Anreiseentfernung und der Beantwortung der Frage nach der Mitgliedschaft im Verein wird die Höhe des Eintrittsgeldes ausgegeben.
Ist ein Mitglied jünger als 15 Jahre oder hat es einen Anreiseweg größer als 50 km, so hat es freien Eintritt. Nichtmitglieder zahlen bei gleichen Voraussetzungen die Hälfte. Alle anderen Besucher zahlen voll.

Benutzeroberfläche:




Struktogramm:



Quelltext:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  alter: integer;
  entfernung: double;
  mitglied: boolean;
begin
  alter := StrToInt(Edit1.Text);
  entfernung := StrToFloat(Edit2.Text);
  if RadioGroup1.ItemIndex = 0 then mitglied := true
    else mitglied := false;
  if (((alter < 15) or (entfernung > 50)) and (mitglied =true)) then
  begin
    ShowMessage('Freier Eintritt!');
    Exit;
  end;
  if (((alter < 15) or (entfernung > 50)) and (mitglied = false))then
  begin
    ShowMessage('Halber Eintritt!');
    Exit;
  end;
  if ((alter >= 15) and (entfernung <= 50)) then
  begin
    ShowMessage('Voller Eintritt!');
    Exit;
  end;
end;
```

Programmtest: Testen Sie das Programm mit folgenden Eingaben!

| Alter | Entfernung | Mitglied |
|---|---|---|
| 14 | 1 | Ja |
| 16 | 34 | Nein |
| 46 | 42 | Ja |
| 15 | 50 | Nein |
| 24 | 58 | Nein |