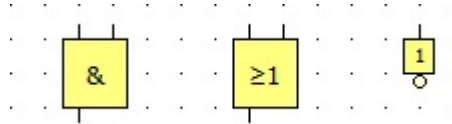


# Grundschrirte in Java

## Logik in Java

Lange ist es her, da haben wir logische Schaltkreise mithilfe von LOCAD betrachtet. Um solche Schaltkreise zu erstellen, gab es drei Grundbausteine: UND-Gatter, ODER-Gatter und Negationspunkt, die hier noch einmal abgebildet sind.



Zur Erinnerung:

- Ein UND-Gatter leitet genau dann Strom weiter, wenn an beiden Eingängen Strom einfließt.
- Ein ODER-Gatter leitet genau dann Strom weiter, wenn an mindestens einem Eingang Strom einfließt.
- Ein Negationspunkt leitet genau dann Strom weiter, wenn kein Strom einfließt.

Auch in Java kann man aus den Grundbausteinen UND, ODER und Negation aufwändige logische Ausdrücke aufbauen. Allerdings arbeiten wir hier nicht mit „Strom an“ und „Strom aus“ sondern mit den sogenannten *Wahrheitswerten* „wahr“ und „falsch“, beziehungsweise „true“ und „false“. Diese Wahrheitswerte kann man ebenso wie Zahlenwerte in Variablen abspeichern und sich deren Werte wieder ausgeben lassen. Diese Variablen müssen dann vom Typ **boolean** sein<sup>1</sup>. Das sieht dann zum Beispiel so aus:

```
boolean meinWahrerWert;  
meinWahrerWert = true;  
  
boolean meinFalscherWert;  
meinFalscherWert = false;  
  
System.out.println(meinWahrerWert);  
System.out.println(meinFalscherWert);
```

**Aufgabe 1.** Erstelle ein Programm, in dem wie oben zwei boolesche Variablen `meinWahrerWert` und `meinFalscherWert` angelegt werden. Die erste soll den Wert **true**, die zweite den Wert **false** haben. Lege dann zwei weitere boolesche Variablen `deinWahrerWert` und `deinFalscherWert` an. Auch hier soll die erste den Wert **true** und die zweite den Wert **false** haben.

Betrachte danach, welche Ausgaben die folgenden Anweisungen liefern:

```
System.out.println(meinWahrerWert && deinWahrerWert);  
System.out.println(meinWahrerWert && deinFalscherWert);  
System.out.println(meinFalscherWert && deinWahrerWert);  
System.out.println(meinFalscherWert && deinFalscherWert);
```

Erkläre damit die Bedeutung der Operation `&&`.

**Aufgabe 2.** Ersetze in den Anweisungen in Deinem Programm aus Aufgabe 1 die Operation `&&` durch `||`. Erkläre dann die Bedeutung der Operation `||`.

*Hinweis: Den senkrechten Strich | erhält man durch AltGr zusammen mit der <-Taste, die unter dem A auf der Tastatur liegt.*

<sup>1</sup>Auf Deutsch würde man sagen, dass dies dann boolesche Variablen sind. Diese Bezeichnung geht zurück auf den Mathematiker und Logiker George Boole.

**Aufgabe 3.** Betrachte nun, welche Ausgaben die Anweisungen

```
System.out.println(!meinWahrerWert);  
System.out.println(!meinFalscherWert);
```

liefern. Erkläre dann die Bedeutung des ! in Java.

**Aufgabe 4.** Ein Informatiker hat sich ein kleines Programm geschrieben, um seine Freizeit zu planen. Jeden Donnerstag muss er nämlich entscheiden, ob er mit seinen Freunden ins Kino oder lieber Joggen geht. Dabei spielen drei Faktoren eine Rolle:

- Seine Freunde haben Zeit.
- Der Film klingt spannend.
- Das Wetter ist gut.

Ob diese drei Faktoren erfüllt sind, gibt er dann jede Woche in drei Variablen an und wartet auf die Ausgabe:

```
public class meinPlaner {  
  
    public static void main(String [] args) {  
  
        boolean freundeHabenZeit;  
        boolean filmSpannend;  
        boolean wetterGut;  
  
        freundeHabenZeit = true;  
        filmSpannend = true;  
        wetterGut = true;  
  
        System.out.println(freundeHabenZeit && filmSpannend && !wetterGut);  
  
    }  
  
}
```

Die Ausgabe **true** bedeutet dann, dass er ins Kino geht, **false** hingegen bedeutet, dass er nicht ins Kino geht, sondern joggt.

Formuliere mit eigenen Worten, in welchen Fällen er ins Kino geht und in welchen er joggt.

**Aufgabe 5.** Wir variieren nun ein wenig das Programm aus Aufgabe 4.

(a) Die Ausgabe des Programms wird nun so abgeändert:

```
System.out.println((freundeHabenZeit || filmSpannend) && !wetterGut);
```

*Hinweis: Die Klammern darf man nicht weglassen, da bei Logik-Rechnungen gilt „Und vor Oder“, so wie beim Rechnen mit Zahlen „Punkt vor Strich“.*

Formuliere mit eigenen Worten, in welchen Fällen der Informatiker nun ins Kino geht. Überprüfe Deine Antwort mithilfe von Java!

(b) Die Ausgabe des Programms wird jetzt so abgeändert:

```
System.out.println(freundeHabenZeit || filmSpannend && !wetterGut);
```

Formuliere wieder mit eigenen Worten, in welchen Fällen der Informatiker nun ins Kino geht. Überprüfe auch hier Deine Antwort mithilfe von Java!

**Zusatzaufgabe 1.** Der Informatiker verfeinert sein Programm nun ein wenig. Er gibt nicht mehr an, ob seine Freunde Zeit haben, sondern wie viele von ihnen Zeit haben. Nun sieht das Ganze so aus:

```
public class meinNeuerPlaner {  
  
    public static void main(String [] args) {  
  
        int freundeHabenZeit;  
        boolean filmSpannend;  
        boolean wetterGut;  
  
        freundeHabenZeit = 2;  
        filmSpannend = true;  
        wetterGut = true;  
  
        System.out.println(freundeHabenZeit >= 2 && filmSpannend ||  
            !wetterGut);  
  
    }  
  
}
```

Formuliere wieder mit eigenen Worten, in welchen Fällen der Informatiker ins Kino geht. Überprüfe auch hier Deine Antwort mithilfe von Java!

**Zusatzaufgabe 2.** Hier ein paar Logik-Knobeleyen. Gegeben seien zwei boolesche Variablen a und b. Die Variable a habe den Wert **true** und die Variable b den Wert **false**.

Ermittle zunächst im Kopf, welche Werte die folgenden Ausdrücke haben:

(a || b) && a

a && b || a

(a || b) && b

a && b && b

a && b || b

!(a || b)

!a || b

a && !b

a || b && a

a && b || !b

a == b

a != b

Anschließend kannst Du Deine Ergebnisse mit Java überprüfen.