

Java

```
boolean istTeilnahmeMöglich(int alter,
    boolean istMitglied, boolean istVerletzt)
{
    if (alter >= 12 && istMitglied == true
        && !istVerletzt)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

UND-Operator
NICHT-Operator

So geht's

- Einfache Bestandteile einer **Bedingung** formulieren, z. B.
 - `alter >= 12`
 - `istMitglied == true`
 - `!istVerletzt`
- Bestandteile mithilfe von **logischen Operatoren** zu einer einzigen **Bedingung** zusammenfassen

Python

```
def istTeilnahmeMöglich(self, alter,
    istMitglied, istVerletzt):
    if (alter >= 12 and istMitglied and
        not istVerletzt):
        return True
    else:
        return False
```

Bedingung ist ein Ausdruck, der ausgewertet WAHR oder FALSCH ergibt.

Bedingungen können mit den **Logischen Operatoren UND, ODER** sowie **NICHT** miteinander zu neuen Bedingungen verknüpfen.

Bedingung:

- Bedingte Anweisung
- Bedingte Wiederholung

A	B	A && B
W	W	W
W	F	F
F	W	F
F	F	F

Heute ist Sonntag (F) UND heute ist Montag (W) => Falsch

A	B	A B
W	W	W
W	F	W
F	W	W
F	F	F

A	!A
W	F
F	W

NICHT Wahr ist Falsch

A oder B

Operator	Bedeutung
<	Kleiner
>	Größer
>=	Echt größer
<=	Echt kleiner
==	gleich

- Anstatt eines Vergleichs (==) wird eine Zuweisung (=) verwendet
- **Bedingung** und bedingte Anweisung verwechselt
- **Bedingung** wertet nie zu WAHR aus
- Für „kleiner gleich“: `=<` anstatt `<=` geschrieben
- falschen **logischen Operator** verwendet

Logische Operatoren:

- Formulieren einfacher Bedingungen
- verknüpfte Bedingungen können ggf. über eine geschachtelte bedingte Anweisung vermieden werden.

Häufige Fehler

Einsatzbereiche

Wahrheitstabellen & Operatoren

