



Ihr habt in dieser Stunde mit Greenfoot weitergearbeitet. Wir haben weitere Übungen zu den bedingten Anweisungen mit logischen Operatoren innerhalb dieser durchgeführt (auf den Blättern die Aufgaben 1-8 ohne 4).

Zusammenfassung

Die logischen Operatoren `&&`, `||` bzw. `!` kann man in Bedingungen beliebig verknüpfen. Obwohl Greenfoot eine feste Reihenfolge „kennt“, würde ich die Ausdrücke so klammern, wie wir das bei der Boole'schen Algebra gelernt haben. Ein Beispiel:

```
if ( wandVorne() && (akkuAufFeld() || !wandRechts()) )
```

```
    {
        dreheRechts();
        schraubeAblegen();
    }
```

```
else
```

```
    {
    }
```

bringt Robby dazu, sich nach rechts zu drehen und eine Schraube abzulegen, wenn folgende Bedingungen erfüllt sind:

```
entweder    ist eine Wand vor ihm UND ein Akku auf seinem Feld
oder        ist eine Wand vor ihm UND KEINE Wand rechts von ihm.
```

Denn eine Wand muss vor ihm sein, denn in der `&&`-Bedingung führt ein „falscher“ Zustand schon zur Unerfüllbarkeit. In der `||`-Bedingung reicht es, wenn eine der (beiden) Bedingungen erfüllt ist.

Durch die Klammerung behält man die Übersicht und weiß genau, was Greenfoot „denken“ soll. Der Ausdruck `„(()))“` ist etwas nervig, aber hier habe ich extra ein Space eingefügt, um die IF-Bedingungsklammer von der logischen Klammer der ODER-Bedingung abzugrenzen, denn...

TABSEN und SPACEN ist alles beim übersichtlichen Programmieren ☺