

## METHODEN

werden **lowerCamelCase** geschrieben. Es gibt zwei unterschiedliche Arten, hier jeweils ein Beispiel:

Verändernde Methode
<pre>void jumpY (int hoehe) {     y = y + hoehe; }</pre> <p>void bedeutet <b>kein Rückgabewert</b> Parameter <b>hoehe</b> vom Typ <b>int</b></p>
<p><b>Aufruf:</b> jumpY (20) ;</p>

Sondierende Methode
<pre>double gibAbstand () {     double abstand = sqrt(x*x+y*y);     return abstand; }</pre> <p>Rückgabewert vom Datentyp <b>double</b></p>
<p><b>Aufruf:</b> double entfernungVomUrsprung = gibAbstand();</p>

### Beachte:

- **Parameter** sind nur im Methodenrumpf, das ist der Teil in den geschweiften Klammern, sichtbar. Globale Variablen sind dagegen in der gesamten Klasse sichtbar. Der **formale Parameter** lebt nur während der Ausführung der Methode (im Methodenspeicher) und stirbt danach.
- Sondierende Methoden müssen das Schlagwort **return** enthalten, außerdem muss der **Datentyp des Rückgabewertes** im Methodenkopf festgelegt werden.

### Aufgaben:

- 1) Schreibe eine Methode `lichtAn(3)`, die eine LED für 3 s einschaltet (die Zahl soll natürlich variabel sein).
- 2) Schreibe eine Methode `getRandNumb(21)`, die eine Integer-Zufallszahl zwischen 0 und 21 zurückgibt. Die „21“ soll hierbei variabel sein. Achte darauf, dass du ein „return“ verwendest.
- 3) Schreibe eine Methode `weihnachtsbaum(5)`, die das Muster unten erzeugt. Die „5“ soll dabei frei wählbar sein. Tipp: verwende verschachtelte Schleifen.w

```
*  
**  
***  
****  
*****
```