Objektorientierte Programmierung
Parameter

Eine besondere Art von Variablen sind *Parametervariablen*. Methoden können keinen, einen oder auch mehrere *Parameter* besitzen. Sie sind lokale Variablen, die nur in der Methode *gültig* sind. Das Besondere ist, dass sie erst beim Aufruf der Methode *initialisiert* (also mit einem Wert belegt) werden.
Du kennst schon einige Methoden mit Parametern:

- `void drehe( String richtung )`
- `boolean huegelVorhanden( String richtung )`

Die *Signatur* der Methoden zeigen an, dass sie beiden eine *Parametervariable* vom Typ `String` übergeben bekommen: `drehe("rechts");`
Bei jedem *Aufruf* der Methode kann dem *Parameter* ein anderer Wert *übergegeben* werden.

Aufgabe 1

Überlegt gemeinsam, welchen Vorteil *Parametervariablen* haben. Vor allem, dass man bei jedem Aufruf andere Werte *übergeben* kann.

Aufgabe 2

Implementiert eine Methode `void fahren( int strecke )`, die den Rover bei jedem Aufruf so viele Felder fahren lässt, wie in der *Parametervariablen* übergeben wird.

Aufgabe 3

Implementiert eine Methode `void fahreRechteck( int kante )`, die die Methode `fahren` von oben benutzt, um den Rover ein Rechteck mit der angegebenen Kantenlänge fahren zu lassen.

Hier den Arbeitauftrag der zweiten Phase einkleben.

Objektorientierte Programmierung
Rückgaben

Eine besondere Art von Variablen sind *Rückgabevariablen*. Methoden können keine oder eine *Rückgabe* besitzen. Sie sind das *Ergebnis* einer Methode, das nach dem Ende der Methode weitergenutzt werden kann. Sie sind also Werte, die aus der Methode „zurück gegeben“ werden.
Du kennst *Rückgaben* schon von den *Anfragen* des Rovers:

- `boolean gesteinVorhanden()`
- `boolean markeVorhanden()`

Die *Signatur* der Methoden zeigen an, dass sie beide als *Rückgabe* einen Wert vom Typ `boolean` *zurück geben*: `boolean wahrheit = gesteinVorhanden();`
Rückgabevariablen haben keinen eigenen *Bezeichner*.

Aufgabe 1

Überlegt gemeinsam, welchen Vorteil *Rückgabevariablen* haben.

Aufgabe 2

Implementiert eine Methode `int zaehleMarken()`, die den Rover die Marken auf dem Feld zählen lässt, und die Anzahl als `Rückgabe` hat.

Hinweis: Ihr könnt in Greenfoot mehrere Marken auf ein Feld legen.

Aufgabe 3

Implementiert eine Methode `int zaehleGesteine()`, die den Rover fünf Felder fahren, und auf dem Weg alle Gesteine analysieren lässt. Die Anzahl Gesteine soll `zurück gegeben` werden.

Hier den Arbeitauftrag der zweiten Phase einkleben.