



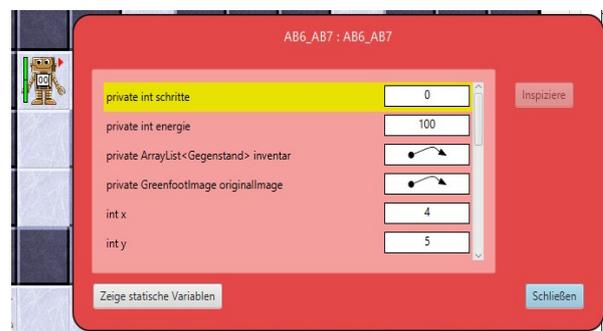
Ein Notfallprogramm für den Roboter... Wir haben Millionen in die Hardware des Roboters und unendlich viel Zeit in die Implementierung der Software und seiner Verbesserungen gesteckt. Daher soll der Roboter ein Notfallprogramm bekommen mit dem er jederzeit den Ausgang wiederfinden kann, wenn er sich in einem zerstörten Kraftwerk befindet und der normale Rückweg plötzlich versperrt ist...

Die Roboter lernen zu zählen ...

ZIEL: Variablen als Speicher für jeweils einen Wert kennen und einsetzen können. Variablen verwenden, um Roboter das Zählen beizubringen.

Aufgaben

1. Erprobe an zwei Robotern **verbraucheEnergie(...)** bzw. **getEnergie()** und kontrolliere mit Inspect (Rechtsmausklick auf den Roboter → Inspizieren) den Zustand der Energie.
2. Bewege und drehe den Roboter und überprüfe, welche Variablen du damit auf welche Werte änderst. Verstehst du das System hinter diesen Werten?
3. Bei den Integer-Attributen kann man die Werte direkt sehen. Bei komplexeren Attributen (z.B. inventar) wird nur ein Pfeil angezeigt. Die Attribute sind komplexer und nicht mit einer einzelnen Zahl auszudrücken. Lass dir durch einen Doppelklick auf den Pfeil hinter inventar anzeigen, was dahinter steckt. Du kannst beispielsweise erkennen, wo der Roboter speichert, wie viele Gegenstände er in seiner Inventarliste hat.
4. **Zählen:** Der Roboter **AB6_AB7** soll eine neue Eigenschaft/Attribut **schritte** bekommen. Deklariere und initialisiere dieses Attribut **schritte**. Erhöhe das Attribut **schritte** in der Methode **einsVorMitZaehlen()** um **1** und gehe ein Schritt nach vorne. Überprüfe im Inspect-Fenster, ob der Roboter seine Schritte korrekt zählt.
5. **Schrittezähler:** Implementiere eine Methode **bisWandMitZaehlen()**, die den Roboter bis zur Wand laufen lässt und dabei die Schritte zählt.
6. **Drehzähler:** Führe ein neues Attribut **drehungen** ein (deklariere und initialisiere das Attribut). Dieses soll bei jeder Rechtsdrehung um **eins erhöht** und bei jeder Linksdrehung um **1 erniedrigt** werden. Führe dazu die beiden Methoden **dreheLinksMitZaehlen()** und **dreheRechtsMitZaehlen()** ein.
7. **Get-Methode:** Schreibe eine Methode **getSchritte()** im Roboter **AB6_AB7**, so dass die Methode den Wert des Attributs zurückgibt. Teste deine Methode.
8. **Get-Methode 2:** Implementiere eine get-Methode für die Anzahl der **Drehungen**. Teste deine get-Methode.

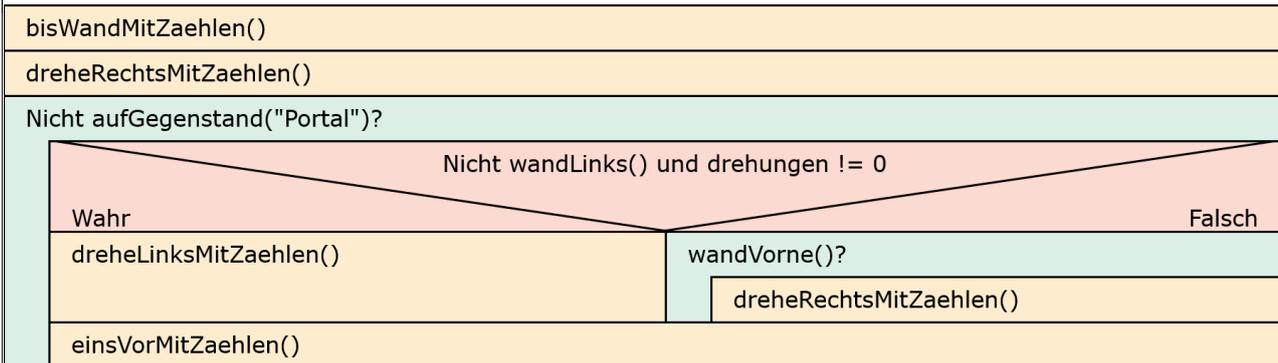




Einsatz 6: Entkommen aus einem unbekanntem Labyrinth

Geht das wirklich? Ja, John Pledge, ein 12-jähriger Junge, hat entdeckt, wie das geht. Wir müssen dazu nur Drehungen zählen können.

Seine Idee war folgendermaßen: Im Prinzip folgt er permanent einer Wand so, dass seine linke Hand die Wand berührt. Da das aber zu Problemen führt, wenn die linke Hand beispielsweise an einer Säule ist, muss man irgendwann loslassen und geradeaus laufen. Das macht Pledge, wenn er sich genauso oft links wie rechtsrum gedreht hat.



Auftrag: Implementiere den Pledge-Algorithmus in der Methode `einsatz6()`. Der Roboter wird dann in einer unbekanntem Umgebung ausgesetzt und muss das Portal finden.

Hinweis: Der grüne Block stellt eine Wiederholung mit `while` und der rote Block eine Fallunterscheidung mit `if-else` dar.



Bildquellen: Die verwendeten Bilder des Roboterszenarios sind alle ohne Bildnachweis verwendbar (selbst gezeichnet, Pixabay Lizenz oder Public Domain). Genaue Nachweise: siehe [bildquellen.html](#).