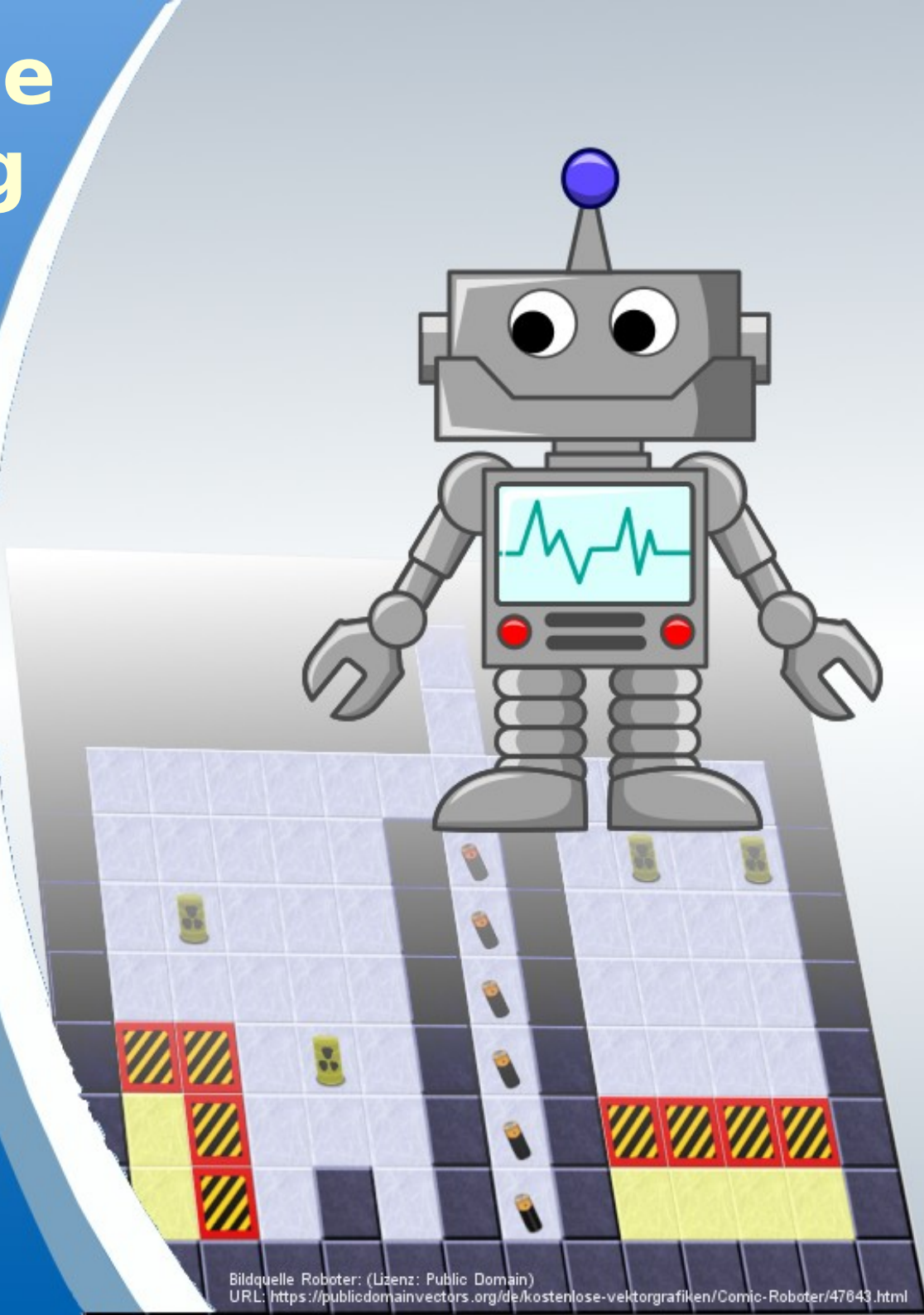
Einführung in die Programmierung

Methoden



Bildquelle Roboter: (Lizenz: Public Domain)
URL: https://publicdomainvectors.org/de/kostenlose-vektorgrafiken/Comic-Roboter/47643.html

INFORMATIK FORTBILDUNG RP KA/FR

Algorithmus

Unter einem Algorithmus verstehen wir eine eindeutige Tätigkeitsbeschreibung, die zum Ziel führt.

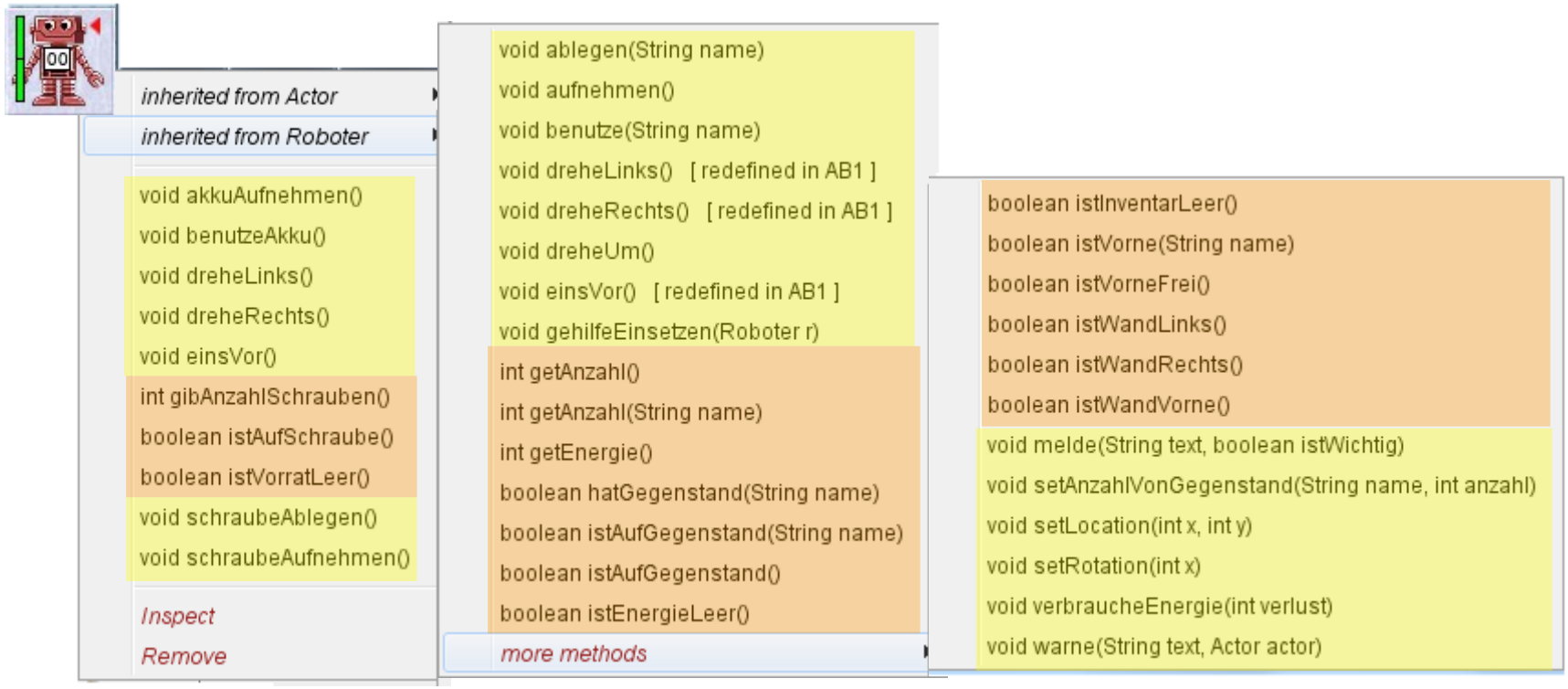
Dabei dürfen nur die zur Verfügung stehenden Befehle und Anfragen verwendet werden.

(nach Wirth und Hromkovic (ETH Zürich))

Zur Verfügung stehende Befehle und Anfragen

```
void einsVor()
void dreheLinks()
void benutzeAkku()
void benutze(String name)
```

```
boolean istVorneFrei()
boolean istAufGegenstand()
boolean istInventarLeer()
int getAnzahl(String name)
```

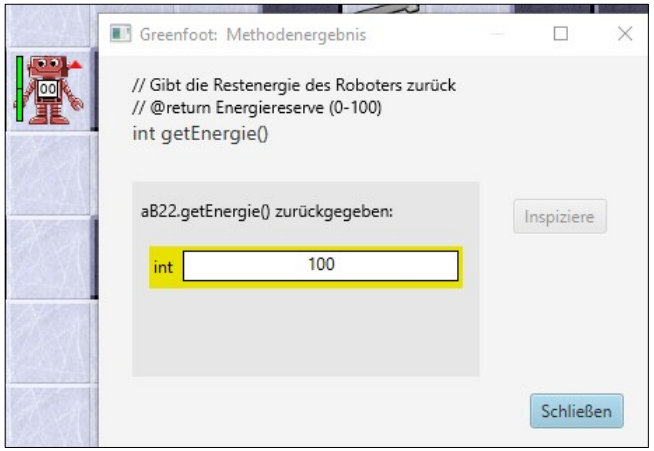


Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html

Fähigkeiten eines Objekts der Klasse Roboter, bei denen **geantwortet** wird, sind z.B.:

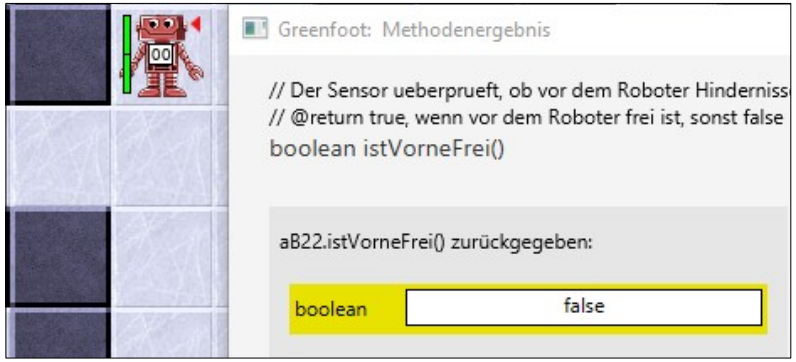
int getEnergie()

Int: Antwort ist eine Zahl



boolean istVorneFrei()

Boolean: Antwort ist true oder false

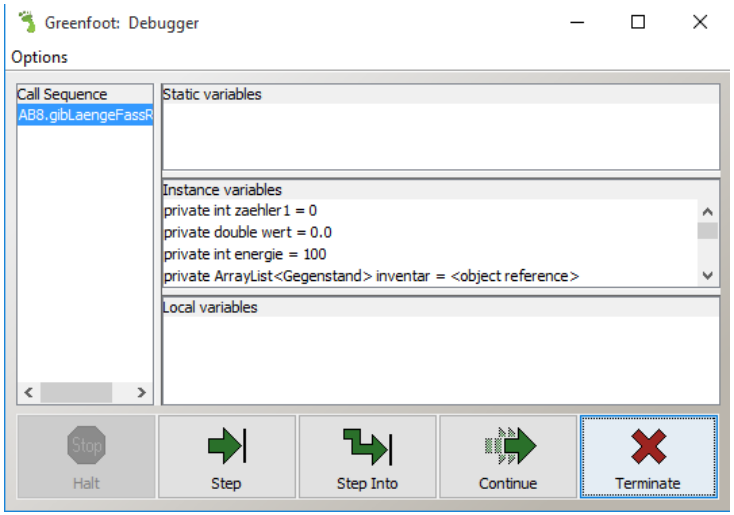


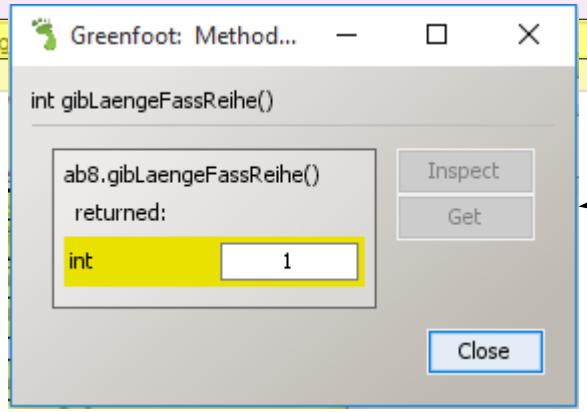
Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html

Programmablauf

```
69  /*#
70   * Aufgabe 7:
71   */
72  public boolean istFassRechts() {
73      dreheRechts();
74      if (istVorne("Fass")) {
75          dreheLinks();
76          return true;
77      }else {
78          dreheLinks();
79          return false;
80      }
81  }
82
83  public int gibLaengeFassReihe() {
84      int laenge;
85      laenge = 0;
86      while(istFassRechts()) {
87          laenge++;
88          einsVor();
89      }
90      return laeng
91  }
```

true
false





1

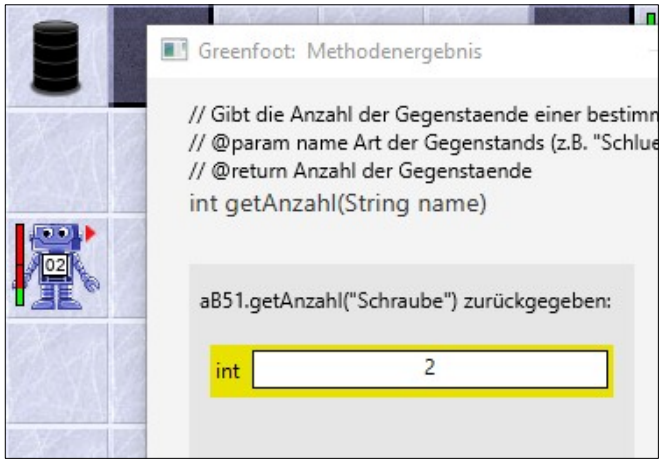
Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html

Methoden mit Parametern

Manche Methoden benötigen Zusatzinformationen, um korrekt arbeiten zu können:

int getAnzahl(String name)

String name: Ein Text ist erforderlich
z.B. getAnzahl ("Schraube");



void setLocation(int x, int y)

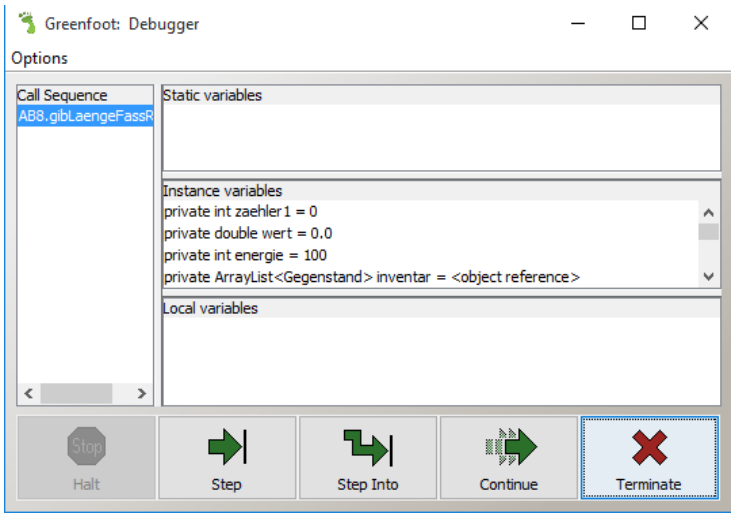
int x, int y: Zwei ganze Zahlen als Koordinaten erforderlich
z.B. setLocation(5,2);

Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html

Programmablauf

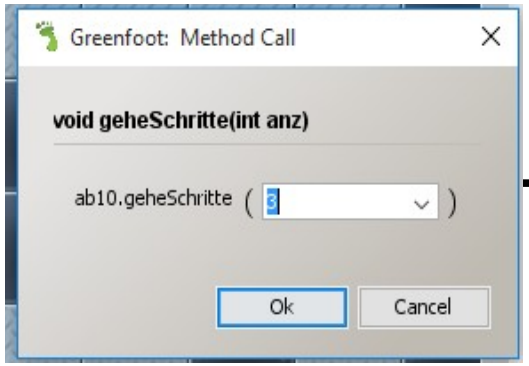
```
/*#
 * Aufgabe 6: Hartes Training
 */
public void gehe3Schritte() {
    geheSchritte(3);
}

public void geheSchritte(int anz) {
    for (int j=0; j<anz; j++) {
        einsVor();
    }
}
```

anz = 3

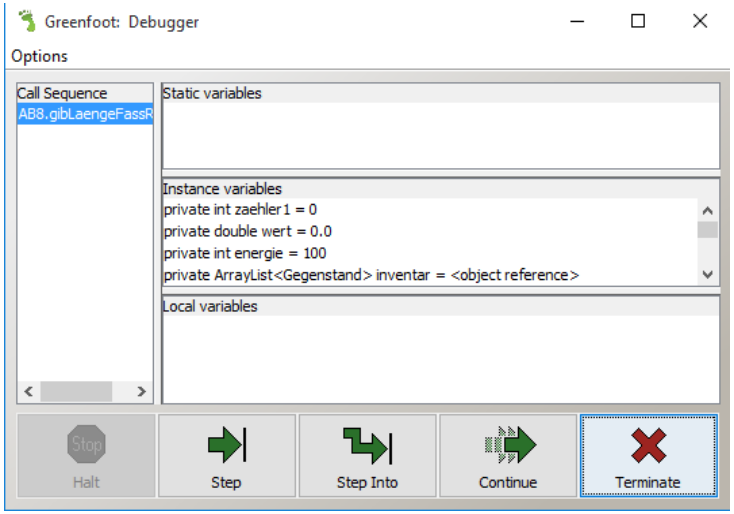


Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html

Programmablauf



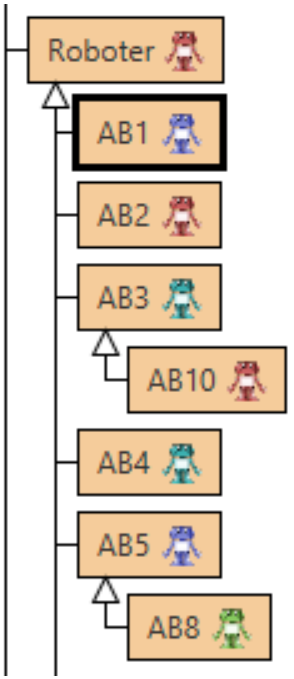
anz = 3




```
/*#
 * Aufgabe 6: Hartes Training
 */
public void gehe3Schritte() {
   geheSchritte(3);
}

public void geheSchritte(int anz) {
   for (int j=0; j<anz; j++) {
      einsVor();
   }
}
```

Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html

Unterklassen erweitern bestehende Klassen.



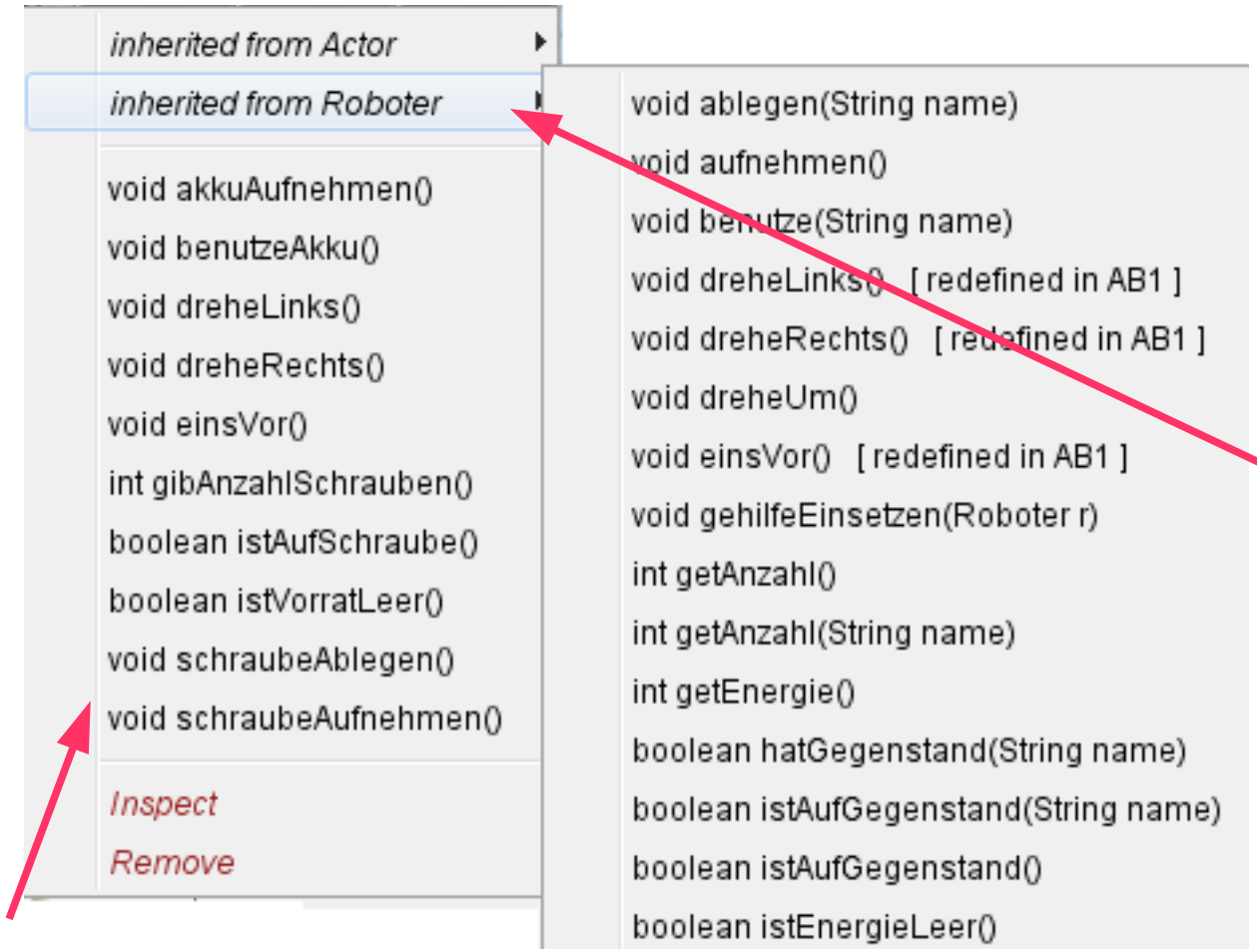
Ein Objekt der Klasse „AB1“ hat die gleichen Eigenschaften und Fähigkeiten wie ein Objekt der Klasse „Roboter“ (A1 erbt von Roboter).

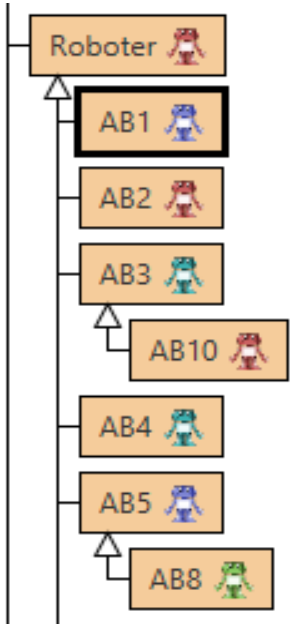
Ein Objekt der Klasse „AB1“ kann die Eigenschaften und Fähigkeiten von „Roboter“ verändern.

Ein Objekt der Klasse „AB1“ kann zusätzlich eigene Eigenschaften und Fähigkeiten haben.

Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html

Unterklassen



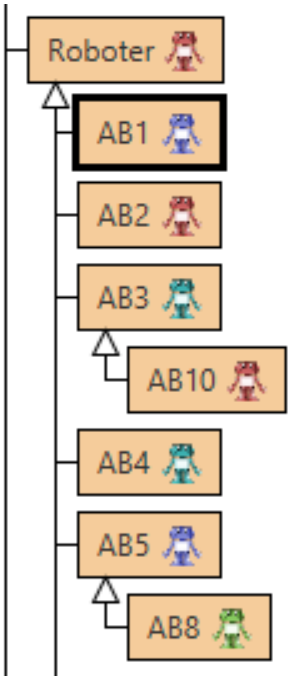


Zusätzlich **erbt** er alle Methoden von Roboter.

Der AB1-Roboter hat z.B. die neuen Fähigkeiten schraubeAblegen() und schraubeAufnehmen()

Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html

Eigene Methoden



Objekte können jederzeit mit neuen Fähigkeiten ausgestattet werden, indem der Klasse eine neue Methode hinzugefügt wird.

Diese Methoden können dann in anderen Methoden verwendet werden.

Komplexe Aufgaben sollten in einfachere Methoden aufgeteilt werden.

Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html