

Name:

Matrikelnummer:

Bearbeitungszeit: 60 min.

Im Paket `robots.egg` sind die `public` Klassen `Thing` und `Robot` gegeben. Die Klasse `Robot` verfügt über den Konstruktor

```
public Robot(int numThings)
```

und die Methoden

```
public void move(),  
public void turnAround(),  
public Set<Thing> thingsCarried(),  
public void pickThing(Thing aThing),  
public void putThing(Thing aThing).
```

Der Konstruktoraufruf `new Robot(numThings)` erzeugt einen Roboter der `numThings` viele Dinge bei sich trägt. Bei Aufruf von `move()` bewegt sich der Roboter einen Schritt nach vorne, bei Aufruf von `turnAround()` dreht sich der Roboter um. Z.B. bewirken die Anweisungen

```
Robot bot = new Robot(0);  
bot.move();  
bot.turnAround();  
bot.move();  
bot.turnAround();
```

dass der Roboter einen Schritt nach vorne macht, umdreht, einen Schritt zurück macht und sich wieder umdreht, sodass er dann wieder am Ausgangspunkt steht und in die ursprüngliche Richtung blickt.

Die Methode `thingsCarried()` liefert die Dinge, die der Roboter gerade bei sich trägt, als `Set` zurück. Der Aufruf `pickThing(aThing)` lässt den Roboter das Ding `aThing` an seiner aktuellen Position aufheben, sofern sich `aThing` dort befindet. Andernfalls liefert die Methode eine Fehlermeldung. Der Aufruf `putThing(aThing)` lässt den Roboter das Ding `aThing` an seiner aktuellen Position ablegen, sofern er es bei sich trägt. Andernfalls liefert die Methode eine Fehlermeldung.

Beim Ausführen der Methode `move()` verliert der Roboter eine zufällige Anzahl der Dinge, die er bei sich trägt. Diese verlorenen Dinge verbleiben an der Ausgangsposition des Roboters, während der Roboter sich einen Schritt nach vorne bewegt. Die Anzahl der verlorenen Dinge kann auch 0 sein, was aber sehr unwahrscheinlich ist, wenn der Roboter viele Dinge bei sich trägt.

Schreiben Sie im Paket `robots` die `public` Unterklasse `SafeRobot` der Klasse `Robot` mit dem Konstruktor

```
public SafeRobot(int numThings),
```

der dem Konstruktor `Robot(numThings)` entspricht.

Überschreiben Sie die Methode `move()` so, dass sich der Roboter nach Ausführen der Methode einen Schritt nach vorne bewegt hat und alle Dinge, die er vor dem Aufruf bei sich hatte, nach wie vor bei sich trägt. Daher muss der Roboter, falls er bei der Bewegung Dinge verloren hat, umkehren um diese aufzuheben, und danach die Bewegung nochmals versuchen.

Es empfiehlt sich, bereits erfolgreich transportierte Dinge abzulegen, bevor verlorene Dinge geholt werden. Die abgelegten Dinge können dann nach erfolgreichem Transport aller Dinge wieder aufgenommen werden.

Zum Testen Ihrer Lösung steht Ihnen die Testklasse `robot.TestSafeRobot` zur Verfügung. Die statische Methode `main(String[] args)` erzeugt einen `SafeRobot` mit 20 Dingen und lässt ihn diese Dinge zwei Schritte transportieren (durch zweimaligen Aufruf von `move()`). Die Bewegung des Roboters wird graphisch dargestellt und der erfolgreiche Transport aller Dinge überprüft. Der Test wird durch Drücken des "Start"-Knopfes begonnen. Nach dem Test soll sich der Roboter am Feld (1,3) befinden.