Übungseinheit 23. - 27. Oktober 2017

Was Sie lernen sollen:

- Schleifen
- Einfache mathematische Funktionen.

Aufgabe 1

Schreiben Sie eine Klasse `Statistiker` mit dem Konstruktor

```
public Statistiker(RobotSE robi)
```

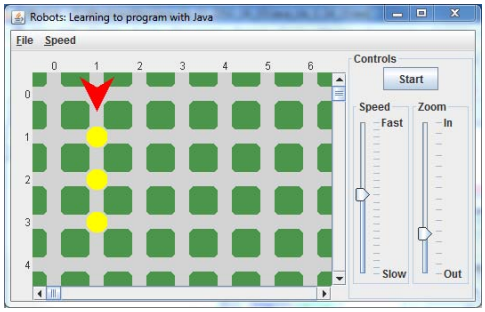
der einen zu steuernden Roboter `robi` in ein Attribut speichert.

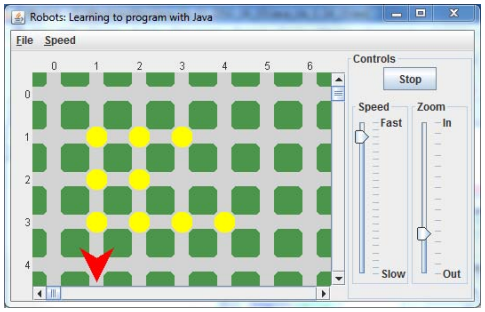
Schreiben Sie in die Klasse `Statistiker` eine Methode

```
public void machBalkendiagramm().
```

Diese Methode soll den Roboter, der vor einer Reihe von Kreuzungen steht, auf denen jeweils ein oder mehrere Dinge liegen (siehe linkes Bild), diese Kreuzungen der Reihe nach besuchen und aus den Dingen, die an den Kreuzungen liegen, ein Balkendiagramm wie im rechten Bild erstellen lassen, bis er eine Zeile ohne Ding erreicht.

Schreiben Sie weiters – nach dem Muster der Beispiele des zweiten Übungsblattes – eine Testklasse `TesteStatistiker` mit einer statischen Methode `static void testeStatistiker()`, die eine City mit einem `Statistiker`, wie im linken Bild angegeben, anlegt und die Methode `machBalkendiagramm()` testet.





Aufgabe 2

Schreiben Sie eine Klasse `IchWillRausRob` mit dem Konstruktor

```
public IchWillRausRob(RobotSE robi)
```

der einen zu steuernden Roboter `robi` in ein Attribut speichert.

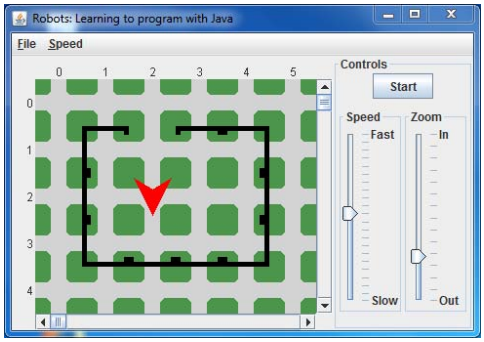
Schreiben Sie in die Klasse `IchWillRausRob` eine Methode

```
public void getOut(),
```

die den in einem Raum mit einem Ausgang platzierten Roboter den Ausgang finden und den Raum verlassen lässt, vorausgesetzt er ist zu Beginn so platziert, dass er nicht auf die Seite des

Raumes mit dem Ausgang blickt (siehe Bild) und der Ausgang nicht in einer Ecke eines Raumes liegt.

Schreiben Sie weiters – nach dem Muster der Beispiele des zweiten Übungsblattes – eine Testklasse `TesteRausRob` mit einer statischen Methode `static void testeRaus()`, die eine City mit einem Raum mit Ausgang erstellt und die Methode `getOut()` testet. Testen Sie mit mindestens drei verschiedenen Ausgangspositionen!



Aufgabe 3

Schreiben Sie eine Klasse `Bergarbeiter` mit dem Konstruktor

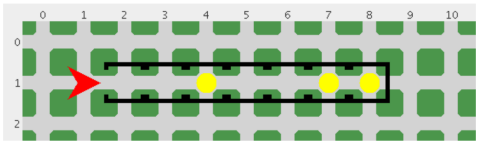
```
public Bergarbeiter(RobotSE robi)
```

der einen zu steuernden Roboter `robi` in ein Attribut speichert.

Ein Bergarbeiter soll über eine Methode

```
public void baueRohstoffAb()
```

verfügen, die den Bergarbeiter alle im Stollen liegenden Dinge zum Eingang bringen lässt.



Schreiben Sie eine Testklasse `TestBergarbeiter` mit einer statischen Methode `static void testAbbau()`, die einen `Stollen`, einen `RobotSE` und einen `Bergarbeiter` erzeugt. Der `Stollen` ist eine `City` und ist von der Lehrveranstaltungsseite herunterzuladen. Der Roboter soll am Anfang des Stollens mit den Koordinaten `(1,1)` und nach Osten schauend erstellt werden, danach soll die Methode `baueRohstoffAb()` ausgeführt werden.

Aufgabe 4

Schreiben Sie eine Klasse `Berechnung` die folgende statische Methode enthält:

```
public static long gibSumme(int anfang, int ende),
```

die die Summe aller geraden Zahlen g, $anfang \leq g \leq ende$ zurückgibt.

Aufgabe 5

Schreiben Sie eine Klasse `Bankomat` mit zwei Konstruktoren

`public Bankomat()`,

`public Bankomat(int anfangsbetrag)`.

Dabei erzeugt `Bankomat()` einen ungefüllten Bankomat und `Bankomat(betrag)` einen Bankomat mit einem verfügbaren Betrag von $betrag$. Mit der Methode

```
public void befuelle(int betrag)
```

kann der Bankomat befüllt werden. Das heißt, dass sich der im Bankomat verfügbare Betrag um den angegebenen `betrag` erhöht. Mit der Methode

```
public int gibVerfuegbarenBetrag()
```

kann der im Bankomat verfügbare Betrag abgefragt werden. Mit der Methode

```
public String behebe(int betrag)
```

kann Geld vom Bankomat behoben werden. Dabei muß der angegebene `betrag` größer als 0 und kleiner oder gleich dem im Bankomat verfügbaren Betrag sein. Ebenso muß der Betrag durch 10 teilbar sein. Der Rückgabe-String soll die Information enthalten, in wie viele Hunderter, Fünfziger und Zehner der `betrag` gestückelt werden kann. Dabei ist jeweils die Maximalzahl an Hundertern und Fünfzigern anzugeben.

Beispiel 1: Wenn Sie für `betrag` den Wert 370 eingeben, sollte die Antwort lauten 'Sie erhalten 3 Hunderter, 1 Fünfziger und 2 Zehner'.

Beispiel 2: Wenn Sie für `betrag` den Wert 230 eingeben, sollte die Antwort lauten 'Sie erhalten 2 Hunderter, 0 Fünfziger und 3 Zehner'.

Beispiel 3: Wenn Sie für `betrag` den Wert 60 eingeben, sollte die Antwort lauten 'Sie erhalten 0 Hunderter, 1 Fünfziger und 1 Zehner'.

Bei erfolgreicher Behebung wird der behobene Betrag nach obigem Schema als String zurückgegeben und der im Bankomat verfügbare Betrag entsprechend reduziert, andernfalls wird 'Ungültiger Betrag' zurückgegeben und der im Bankomat verfügbare Betrag bleibt unverändert.

Aufgabe 6

Schreiben Sie eine Klasse `EinfacheFunktionen` die folgende statische Methoden enthält:

```
public static long faktorielle(int n),
```

mit der Sie $n!$ berechnen können, sowie eine Methode

```
public static double eHochX(double x)
```

zur näherungsweisen Berechnung von e^x. Die Funktion e^x kann durch die Reihe $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$ angenähert werden. Die Methode `eHochX` soll diese Reihe bis zum neunten Glied auswerten. Die Verwendung der Methode `faktorielle` wird dabei hilfreich sein.