Technische Universität Darmstadt
Fachbereich Informatik
Prof. Dr. Johannes Fürnkranz

Allgemeine Informatik 2 im SS 2007

Übungsblatt 4

Bearbeitungszeit: 23.05. bis 29.05.2006

Aufgabe 1: Die Rechteck-Welt

Die Bewohner einer Rechteck-Welt sind – natürlich – Rechtecke. Sie charakterisieren sich durch ihre Positionskoordinaten in der Rechteckwelt (einem zweidimensionalen Koordinatensystem) sowie ihrer Breite und Höhe. Allen gemeinsam ist, dass die Seitenlängen ganzzahlig und nicht-negativ, sowie die Seiten parallel zu den Koordinatenachsen sind.

a) Klasse: Starten Sie BlueJ und definieren Sie eine Klasse **`Rectangle`**, die Rechtecke repräsentiert. Löschen Sie die vordefinierten Inhalte der Klasse wie in den letzten Übungen auch.

b) Konstruktor: Um ein Rechteck zu erzeugen und zu initialisieren, benötigen Sie einen Konstruktor. Definieren Sie in der Klasse **`Rectangle`** einen Konstruktor, der das neue Objekt mit den Koordinaten der linken, unteren Ecke eines Rechtecks und dessen Höhe und Breite initialisiert. Sie benötigen Variablen auf der Klassenebene, damit Sie die im Konstruktor übergebenen Werte speichern können.

Wenn sich in der Rechteckwelt Rechtecke begegnen, wollen sie sich näher kennen lernen. Oft fragt man da nach den charakterisierenden Eigenschaften.

c) Methoden: Schreiben Sie Methoden **`getWidth`**, **`getHeight`**, **`getLeft`**, **`getRight`**, **`getBottom`** und **`getTop`**, die entsprechend ihres Namens die Breite, Höhe und die Begrenzungen (links, rechts, unten, oben) eines Rechtecks zurückgeben.

d) Methoden: Schließlich wollen Sie ganz besonders wissen, ob das Rechteck ein Quadrat ist. Die Antwort soll der Aufruf der Methode **`isSquare()`** mit dem Rückgabetyp **`boolean`** liefern.

Auch in der Rechteckwelt finden sich Rechtecke, die in einer Beziehung zueinander stehen. Dazu fragt man ein Rechteck, ob es in einer bestimmten Beziehung zu einem anderen Rechteck steht.

e) Methoden: Definieren und implementieren Sie folgende Methoden:

- **`boolean isContained(Rectangle r)`**
  liefert **`true`**, wenn jeder Punkt dieses Rechtecks auch Punkt von r ist, und ansonsten **`false`**.

- **`boolean isDisjoint (Rectangle r)`**
  liefert **`true`**, wenn dieses Rechteck und r keinen gemeinsamen Punkt haben, und ansonsten **`false`**.

f) optionale Methoden: Sie können sich gerne auch noch weitere Methoden ausdenken, die etwas über die Beziehungen zwischen Rechtecken herausfinden.

g) weitere Klasse: Schreiben Sie außerdem eine weitere Java-Klasse mit ausschließlich einer **`main`**-Methode, das die Rechteck-Klasse benutzt und mehrere Rechtecke erzeugt und miteinander vergleicht.

Aufgabe 2: Straßenbahn

Sicherlich sind Sie in Darmstadt schon Straßenbahn gefahren. Eine Straßenbahn hat eine Liniennummer. Sie kann fahren und wieder anhalten und es können Passagiere ein- und aussteigen (aber nur während des Haltevorgangs).

Implementieren Sie eine Java-Klasse, die die genannten Eigenschaften und Funktionen einer Straßenbahn simuliert. Gehen Sie dabei wie folgt vor:

a) Erstellen Sie eine Klasse **`Tram`**. Löschen Sie bitte wieder die vordefinierten Inhalte.
b) Definieren Sie in der Klasse folgende Variablen, die den Zustand einer Straßenbahn charakterisieren:
   - **`boolean driveMode`**
     `true`, falls Straßenbahn fährt, ansonsten **`false`**
   - **`int line`**
     Die Liniennummer der Straßenbahn
   - **`int passenger`**
     Die Anzahl der Passagiere, die in der Straßenbahn sind.

c) Die Klasse soll einen Konstruktor besitzen, der die Liniennummer als Argument bekommt, und die Variablen mit Anfangswerten initialisiert. Zu Beginn steht natürlich die Straßenbahn, und es befinden sich keine Passagiere in der Bahn.

d) Schreiben Sie folgende Methoden und implementieren Sie die folgenden Funktionalitäten
   - **`void drive ()`**
     Die Straßenbahn wird in den Fahrmodus gesetzt.
   - **`void halt ()`**
     Die Straßenbahn hält an.
   - **`void getIn (int n)`**
     Es steigen **`n`** Passagiere ein.
   - **`void getOut(int n)`**
     Es steigen **`n`** Passagiere aus.

e) Geben Sie auf dem Bildschirm Informationen aus, wenn ihre Straßenbahn fährt oder hält, oder wie viele Passagiere ein- bzw. aussteigen.

f) Geben Sie Fehlermeldungen aus, wenn im aktuellen Zustand nicht sinnvolle Methoden aufgerufen werden. Es sollte nicht möglich sein, dass Passagiere ein- oder aussteigen, während die Straßenbahn fährt.

g) Verwenden Sie zum Testen die bereitgestellte Klasse **`TramTest`**. Importieren Sie diese in BlueJ. Wenn Sie die darin enthaltene **`main`**-Methode ausführen, dann sollte eine Ausgabe erscheinen, wie nachfolgend abgebildet:

```
1  Linie 1: Betriebsbereit.
2  Linie 7: Betriebsbereit.
3  Linie 1: Faehrt.
4  Linie 7: Es sind 5 Passagiere eingestiegen.
5  Linie 7: Faehrt.
6  Linie 7: Haelt.
7  Linie 7: Es sind 4 Passagiere ausgestiegen.
8  Linie 1: Einsteigen nicht möglich. Bahn faehrt.
9  Linie 1: Haelt.
10 Linie 1: So viele Passagiere sind nicht in der Bahn. Vorgang abgebrochen.
```