Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Einführung in die Informatik
Sommersemester 2012

Prof. Dr. Wolfram Burgard
PD Dr. Cyrill Stachniss
Dipl. Ing. Jörg Röwekämper
Dipl. Inf. Markus Kuderer

Übungsblatt 12

Abgabe bis Freitag, 27.07.2012, 12:00 Uhr

Hinweis:

Aufgaben immer per E-Mail (eine E-Mail pro Blatt und Gruppe) an den zuständigen Tutor schicken (Bei Programmieraufgaben Java Quellcode und eventuell benötigte Datendateien).

Aufgabe 12.1

1. Implementieren Sie ein Interface `GeometricObject`, das die Methoden `double area()` und `double circumference()` besitzt.
2. Ändern Sie die Klasse `Circle` von Übungsblatt 5 (siehe Vorlesungshomepage) so ab, dass sie das Interface `GeometricObject` implementiert.
3. Vervollständigen Sie folgende Klasse `Rectangle`, die ebenfalls das Interface `GeometricObject` implementiert.

```
class Rectangle ... {
  public Rectangle(double length, double width) { ... }

  public double area() { ... }
  public double circumference() { ... }

  protected double length;
  protected double width;
}
```

4. Implementieren Sie eine Klasse `Square`, die von `Rectangle` abgeleitet ist. Die Seitenlänge des Quadrates soll im Konstruktor gesetzt werden.
5. Könnten die Instanzvariablen `length` und `width` auch als `private` deklariert werden und welche Implikationen hätte dies auf die Klasse `Square`?
6. Implementieren Sie die Klasse `GeometricObjectVector`, die mehrere geometrische Objekte verwalten soll. Diese Klasse soll als Instanzvariable einen Vektor `Vector<GeometricObject>` besitzen und folgende Methoden zur Verfügung stellen:

   (a) `void add(GeometricObject o)`
   fügt ein Objekt in den Vektor ein.

(b) `double areaSum()`
berechnet die Summe des Flächeninhalts aller Objekte in dem Vektor.

(c) `double circumferenceSum()`
berechnet die Summe des Umfangs aller Objekte in dem Vektor.

(d) `GeometricObject maxArea()`
gibt das Objekt mit dem größten Flächeninhalt zurück.

Aufgabe 12.2

In der Vorlesung wurde die einfach verkettete Liste `SingleLinkedList` mit dem Iterator `SingleLinkedListIterator` eingeführt. Sie finden die Implementierung auf Vorlesungshomepage unter "Beispielprogramme".

1. Implementieren Sie für den Iterator die Methode `void remove()`. Diese Methode löscht das Element der Liste, das als letztes von `next()` zurückgegeben wurde. Ist es möglich, diese Operation in Konstantzeit ($O(1)$) auszuführen?

2. Implementieren Sie eine Methode
`swapContent(Node_SLL<G> p1, Node_SLL<G> p2)`,
welche den **Inhalt** der Knoten `p1` und `p2` vertauscht.
Welche Komplexität hat diese Methode?

3. Implementieren Sie eine Methode
`swapNode(Node_SLL<G> p1, Node_SLL<G> p2)`,
welche die beiden **Knotenelemente** vertauscht. Dadurch soll sich die Reihenfolge der durch `p1` und `p2` referenzierten Elemente in der Liste ändern.
Welche Komplexität hat diese Methode?

4. In welchen Situationen haben die beiden Methoden einen unterschiedlichen Effekt?

Aufgabe 12.3

Im PacMan Spiel gibt es verschiedene Objekte, die miteinander interagieren, z.B. die Monster und die Pillen. Die Objekte haben verschiedene Gemeinsamkeiten. So besitzt jedes von ihnen einen Kreis, der es umgibt, für die Kollisionserkennung sowie seine Position.

Schreibe Sie eine abstrakte Klasse `Entity` welche die Membervariablen

1. `Circle boundingCircle` und

2. `BufferedImage image`

besitzt. Des Weiteren soll die Klasse Set- und Get-Methoden für Membervariablaien enhalten und die Methode `public void draw(Graphics g)`, die am Mittelpunkt des Circles das BufferedImage zentriert zeichnet.

Lassen Sie die Klasse `Creature` von der Klasse `Entity` erben. Entfernen Sie nicht die bereits implementieren Methoden der Klasse `Entity`.

Schreiben Sie eine Klasse `Pille`, die von `Entity` erbt.