Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Einführung in die Informatik
Sommersemester 2013

PD Dr. Cyrill Stachniss

Dr. Rainer Kümmerle

Übungsblatt 8

Abgabe bis Freitag, 21.06.2013, 10:00 Uhr

Hinweis:

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

Aufgabe 8.1

1. Laden Sie das Beispielprojekt `MyProject`[1] von der Vorlesungsseite herunter. Kompilieren Sie die in `MyProject` enthaltenen Beispielprogramme mit Hilfe des `ant`-Buildsystemes (Siehe Foliensatz Tools).

2. Erstellen Sie Ihr eigenes Projekt `Info1Exercises`. Übernehmen Sie dafür die Struktur des Beispielprojektes. Fügen Sie die Klasse `Set`, die Sie auf der Vorlesungsseite herunterladen können, in das `src`- Verzeichnis ein und kompilieren Sie Ihr Projekt. Betrachten Sie die Ausgabe des `ant`-Buildsystemes und überprüfen Sie, ob alle Codekonventionen eingehalten wurden. Korrigieren Sie den Quelltext entsprechend der Meldungen.

3. Schreiben Sie eine Klasse `SetTest`, die `JUnit`-Tests für die Methoden `addElement` und `contains` der Klasse `Set` enthält. Testen Sie hierbei insbesondere, dass die Eigenschaften einer Menge eingehalten werden.

Aufgabe 8.2

Betrachten Sie den Auszug der Klasse `GenericTuple`, die ein Tupel aus drei Objekten darstellt.

```
public class GenericTuple<A,B,C> {
  public void setA( ... ) { ... }
  public void setB( ... ) { ... }
  public void setC( ... ) { ... }

  public ... getA(){ ... }
  public ... getB(){ ... }
  public ... getC(){ ... }

  public String toString() {
    ...
  }
```

[1] http://ais.informatik.uni-freiburg.de/teaching/ss13/info/material/MyProject.zip

```
  A a;
  B b;
  C c;
}
```

1. Vervollständigen Sie die `set` - und `get`-Methoden der Klasse `GenericTuple`.

2. Vervollständigen Sie die `toString` Methoden der Klasse `GenericTuple`, so dass sie einen String der Form "$(\cdots,\cdots,\cdots)$" ausgibt.
   Hinweis: Verwenden Sie die `toString`-Methoden der drei Member-Variablen.

3. Implementieren Sie eine Klasse `GenericTupleVector<A,B,C>`, die einen Vektor aus `GenericTuple<A,B,C>`-Elementen enthält. Der Vektor soll hierbei mit Hilfe der Klasse `ArrayList` repräsentiert werden.

4. Implementieren Sie für die Klasse `GenericTupleVector<A,B,C>` eine Methode `int getSize()`, die die Anzahl der Elemente im Vektor zurückgibt.

5. Implementieren Sie für die Klasse `GenericTupleVector<A,B,C>` eine Methode `addTuple(GenericTuple<A,B,C> tuple)`, die ein Tupel in den Vektor einfügt.

6. Implementieren Sie für die Klasse `GenericTupleVector<A,B,C>` eine Methode `GenericTupleVector<A,B,C> compareB(B b)`, die einen Vektor zurückgibt, der alle Tupel `t` enthält, für die gilt `b.equals(t.getB())`

7. Auf der Vorlesungshomepage finden Sie eine `JUnit`−Klasse `GenericTupleVectorTest.java`. Nutzen Sie diese, um ihre Klassen zu testen.