hochschule mannheim

Fakultät für Informatik
Prof. Dr. Peter Knauber

Bachelor-Studiengang Informatik
Übungen zur Vorlesung “Grundlagen der Informatik” (GDI), WS 2010/2011

Übungsblatt 6: Interfaces, Klassen und Objekte

Ausgabe am: 18.11.2010
Abgabe am: 3.12.2010

Aufgabe 1: Bibliothekssystem **10 Punkte**

Erstellen Sie in Java ein Interface für eine Bibliothek bzw. ein Bibliotheksprogramm. Überlegen Sie, welche Aufgaben anstehen, welche Parameter jeweils dafür benötigt werden (z.B. Bücher, Euro-Beträge etc.) und welche Rückgaben sinnvoll sind. Tipp:

Schreiben Sie eine *main*-Methode, die Ihre Interface-Methoden aufruft und versetzen Sie sich dabei in die Rolle desjenigen, der die Aufgabe tatsächlich erledigt haben möchte. Überlegen Sie dann, welche Parameter notwendig sind und welches Ergebnis Sie erwarten. Denken Sie an Sonderfälle.

Kommentieren Sie alle Methoden gründlich mittels JavaDoc.

Definieren Sie (leere) Interfaces für alle weiteren Datentypen, die Sie für Ihre Lösung benötigen.

Aufgabe 2: U-Bahn-Fahrkarten **12 + 3 = 15 Punkte**

Erstellen Sie ein Interface *Fahrkartensystem* für ein elektronisches U-Bahn Fahrkartensystem. Natürlich soll in der U-Bahn niemand mit Bargeld hantieren müssen und die Kosten berechnen sich nach der genauen Fahrstrecke. Überlegen Sie, welche Aufgaben anstehen, welche Parameter jeweils dafür benötigt werden (z.B. Streckenlänge, Euro-Beträge etc.) und welche Rückgaben sinnvoll sind. Kommentieren Sie alle Methoden gründlich mittels JavaDoc; denken Sie dabei an Sonderfälle.

Ihr Fahrkartensystem soll zumindest folgende Aufgaben erledigen können:

- einen neuen Kunden erstellen
- ein Kundenkonto aufladen
- den Kontostand abfragen
- ein Kundenkonto auflösen
- Kunde möchte einsteigen (wird an der Einstiegshaltestelle gehindert wenn kein Geld auf der Karte)
- Kunde möchte aussteigen (abbuchen anhand von Fahrstrecke, hindern bei Fehler oder zu wenig Geld)

Definieren Sie (leere) Interfaces für alle weiteren Datentypen, die Sie für Ihre Lösung benötigen.

Aufgabe 3: Große Zahlen **18 Punkte**

Schreiben Sie eine Klasse *BigInteger*, welche das Rechnen mit sehr großen natürlichen Zahlen (inklusive 0) ermöglicht. Die Klasse soll eine Zahl ziffernweise in einem Array vom Typ *int* speichern. Sie soll über einen parametrisierten Konstruktor verfügen, der eine Zahl als String entgegen nimmt. Besteht der String lediglich aus Ziffern, wird das Array mit entsprechender Länge initialisiert und die Zahl ziffernweise im Array gespeichert. Das Array soll dabei lediglich so groß wie unbedingt notwendig sein. Bei ungültigem Parameter soll das Array die Zahl 0 repräsentieren. Die Klasse soll weiter über folgende Methoden verfügen:

- void add(BigInteger number)
  Erhöht die Zahl um die als Parameter übergebene Zahl.
  Hinweis: Vergegenwärtigen Sie sich bei der Implementierung die schriftliche Addition.
- String toString()
  Liefert die Zahl als String.
- int[] getDigits()
  Liefert das Array in dem die Ziffern gespeichert wurden
- int length()
  Liefert die Anzahl der Ziffern der Zahl.

Erstellen Sie eine weitere Klasse, die in einer *main*-Methode *BigInteger*-Objekte erzeugt und mit Testaufrufen die Funktionalität von deren Methoden unter Beweis stellt.

Vergessen Sie nicht, Ihr Programm zu kommentieren!

Hinweise

- Die Aufgaben sind in Eclipse zu bearbeiten. Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket (engl. Package) namens *uebung06* an.
- Von allen Aufgaben sind Listings abzugeben, *keine* Testläufe.
- Erlaubt sind *MakeItSimple*-Funktionen (keine nicht besprochene Funktionalität aus der Java Standard Bibliothek) und das bisher erworbene Wissen aus den GDI-Vorlesungen. Zusätzliche eigene Hilfsfunktionen (keine fremden oder externen) sind ausdrücklich erlaubt.
- In den Laborstunden soll Ihre Lösung zu Aufgabe 3 automatisch getestet werden. Damit Sie vorab prüfen können, ob Ihr jeweiliges Programm äußerlich korrekt ist (was nicht heißt, dass es korrekt funktioniert!), finden Sie im Wiki dafür ein JUnit-Testprogramm.