

Bachelor-Studiengang Informatik
Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2011/2012

Übungsblatt 10: Interfaces und Klassen

Ausgabe am: 2.12.2011
keine Abgabe, nur Betreuung**Aufgabe 1: Ein Bibliotheksprogramm** **0 Punkte**

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

Schreiben Sie zuerst 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. Erstellen Sie geeignete Interfaces oder Klassen für alle Datentypen, die Sie brauchen können (z.B. für Bücher, Euro-Beträge, Datumsangaben etc.).

Verwenden Sie nur deutsche Namen für Klassen, Methoden, Parameter etc.

Kommentieren Sie alle Interfaces, Klassen und Methoden mittels Javadoc.

Aufgabe 2: Container **0 Punkte**

Sie sollen eine Software-Repräsentation für Container implementieren, wie sie beim Transport von Gütern verwendet werden:

- Ein Container verfügt immer über Breite, Höhe, Länge in Metern, ein daraus resultierendes Volumen sowie in manchen Fällen über ein maximal zulässiges Ladegewicht in Tonnen.
- Für einen konkreten Einsatz wird der Container mit einem Inhalt versehen und er bekommt ein Transportziel mit der zugehörigen Entfernung angegeben.
- Die Kosten für den Transport belaufen sich pauschal auf 2 Euro pro Kubikmeter und Transportkilometer.
- Das Volumen und die Kosten für einen konkreten Transportweg sollen aus den vorliegenden Angaben automatisch berechnet werden.
- Wenn ein Container seinen Zielort erreicht hat, wird er wieder entladen.
- Es muss möglich sein zu erfahren, ob ein Container gerade verwendet wird (Inhalt etc. angegeben) oder nicht.
- Es soll eine Methode *toString()* geben, die alle zu einem Container vorliegenden Angaben als Klartext in einem String liefert.

- a) Erstellen Sie eine *Container*-Klasse mit einer *main*-Methode, in der Sie die Konstruktoren und Methoden, die die Klasse bekommen soll, aufrufen (kein Menü erforderlich, nur konkrete Aufrufe).
- b) Erstellen Sie geeignete Konstruktoren für Ihre Klasse *Container* und dokumentieren Sie sie mittels Javadoc.

- c) Erstellen Sie geeignete Methoden für die verschiedenen Aufgaben der Container und dokumentieren Sie sie mittels Javadoc.
- d) Erstellen Sie geeignete Felder für alle Angaben und sorgen Sie dafür, dass diese von den Konstruktoren mit passenden Werten belegt werden.

Halten Sie eine einheitliche Reihenfolge der gemeinsamen Parameter in den verschiedenen Parameterlisten ein.

Aufgabe 3: Große Zahlen**0 Punkte**

Um das Rechnen mit sehr großen Zahlen zu ermöglichen schreiben Sie eine Klasse *BigInt* welche dies für Natürliche Zahlen (inklusive 0) ermöglicht. Die Klasse soll dabei eine Zahl ziffernweise in einem Feld vom Typ *int* speichern. Der Konstruktor der Klasse soll eine Zahl als String entgegennehmen können. Besteht der String lediglich aus Ziffern wird das Feld mit entsprechender Länge initialisiert und die Zahl ziffernweise im Feld gespeichert. Das Feld soll dabei lediglich so groß wie unbedingt notwendig sein. Bei ungültigem Parameter soll das Feld die Zahl "0" repräsentieren. Die Klasse soll weiter über folgende Methoden verfügen:

- *void add(BigInt number)*
Addiert eine andere Zahl vom Typ *BigInt* zu der Zahl. Vergegenwärtigen Sie sich bei der Implementierung die schriftliche Addition.
- *String toString()*
Liefert die Zahl als String.
- *int[] getDigits()*
Liefert das Feld in dem die Ziffern gespeichert wurden.
- *int length()*
Liefert die Anzahl der Ziffern der Zahl.

- a) Erstellen Sie ein Interface für diese Methoden mit geeigneten Javadoc-Kommentaren.
- b) Erstellen Sie eine Klasse mit geeignetem Konstruktor, die dieses Interface implementiert.
- c) Erstellen Sie eine weitere Klasse mit einer *main*-Methode, um Ihre Implementierung zu testen.

Hinweise

- Die Bearbeitung dieses Übungsblattes erfolgt auf freiwilliger Basis: Die Aufgaben vertiefen den in der Vorlesung behandelten Stoff, sie müssen aber nicht bearbeitet und abgegeben werden. Nutzen Sie die Aufgaben, um sich selbst zu überprüfen, inwieweit Sie den Stoff verstanden haben und nutzen Sie die Laborstunden, um eventuelle Fragen zu klären, sich Hilfestellung zu holen etc.
- Wenn Sie die Aufgaben nicht bearbeiten, nutzen Sie die Laborstunden ohne Testat dazu, sich Hilfestellung zum regulären Übungsblatt 9 zu holen.
- Die Aufgaben sind in Eclipse zu bearbeiten, legen Sie dafür ein Package *uebung010* an.