

Bachelor-Studiengang Informatik
Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2010/2011Übungsblatt 8: Klassen, Konstruktoren, Objekte
Abgabe am: 9.12.2010
Abgabe am: 10.1.2011¹**Aufgabe 1: Container** **6+3+3 = 12 Punkte**

Erstellen Sie eine Klasse *Container*, die einen Container repräsentiert, wie er beim Transport von Gütern verwendet wird. Ihre Klasse soll folgende Felder besitzen:

- int length
- int width
- int height
- int weight
- String content
- int volume
- int fees

Ein Container verfügt immer über eine Breite, Höhe, Länge und ein Volumen. Ein weiteres Feld stellt die anfallenden Kosten für den Transport dar. Die Kosten für den Transport belaufen sich auf 2 Euro pro Kubikmeter und Transportkilometer. Kosten und Volumen sollen automatisch berechnet werden. In einigen Fällen muss die Möglichkeit bestehen, Gewicht und Inhalt zu vermerken. Deshalb soll Ihre Klasse neben dem Konstruktor mit der minimalen Parameterliste über zwei zusätzliche Konstruktoren verfügen, die es ermöglichen, zusätzlich das Gewicht bzw. Gewicht und Inhalt anzugeben. Sind Gewicht oder Inhalt nicht bekannt soll 0 bzw. "n/a" eingetragen sein. Weitere Randbedingungen:

- Halten Sie eine einheitliche Reihenfolge der gemeinsamen Parameter in den verschiedenen Parameterlisten ein.
- Vermeiden Sie in dieser Aufgabe redundanten Code!

Aufgabenstellung:

- Implementieren Sie die Klasse *Container*. Dokumentieren Sie alle Konstruktoren mittels JavaDoc.
- Erstellen Sie eine *main*-Methode in Ihrer *Container*-Klasse, um Ihr Programm zu testen.
- Diskutieren Sie schriftlich in Form eines JavaDoc-Kommentars zum *Volumen*-Feld: Ist es sinnvoll, das Volumen durch ein Feld darzustellen? Welche Alternative gibt es? Was spricht für die eine, was für die andere Lösung?

1. Senden Sie Ihre Listings bis spätestens zum 10. Januar 2011 um 10 Uhr an Ihren Tutor für dieses Übungsblatt. Erfragen Sie zuvor *in einer Übungsstunde*, wer sie betreuen wird.

Aufgabe 2: Datumsberechnungen**16+5+3 = 24 Punkte**

Auf der Vorlesungshomepage finden Sie ein Interface *Date*, das definiert, welche Operationen konkrete Datumsangaben (mit Tag, Monat und Jahr) zur Verfügung stellen. Dazu gehören zum Beispiel Operationen zum Speichern des *Date*-Zustands in eine Datei und zum Laden der geschriebenen Daten aus einer Datei. Aufgabenstellung:

- Implementieren Sie dieses Interface in einer Klasse *DateImpl* mit einem Konstruktor, der alle angegebenen Daten als Parameter akzeptiert.
- Überlegen Sie, wie Sie mit Sonderfällen wie zum Beispiel ungültigen Datumsangaben umgehen in Form von JavaDoc-Kommentaren bei den betroffenen Methoden/Konstruktoren.
- Erstellen Sie eine *main*-Methode in Ihrer *DateImpl*-Klasse, um Ihr Programm zu testen.

Es ist Ihnen überlassen, wie Sie die Methoden implementieren (z.B. welche Hilfsmethoden Sie entwickeln und verwenden möchten), aber Sie dürfen keine fremden Bibliotheken verwenden.

Hinweise

- Die Aufgaben sind in Eclipse zu bearbeiten. Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket (engl. Package) namens *uebung08* 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 2 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.

Es wird nächste oder übernächste Woche ein freiwilliges Übungsblatt geben, mit dem (etwa 25) Zusatzpunkte erworben werden können. Die zugehörigen Testate finden zusammen mit den Testaten für dieses Übungsblatt 8 in der ersten Januarwoche statt.