Pflichtaufgabe 4 – Mitfahrzentrale

Software für ein Mitfahrdienst

Entwickeln Sie eine Software für einen Mitfahrdienst ("Mitfahrerzentrale" o.ä.) Die Software ist OpenSource und wird von einem zentralen Service verwaltet. Jedes Unternehmen kann ein in das interne Firmensystem mit Hilfe Ihrer Programmbibliothek an dem Mitfahrerdienst teilnehmen und beispielsweise Ihre Mitarbeiter und Mitarbeiterinnen nahelegen diesen zu nutzen, um umweltbelastende Inlandsflüge zu minimieren.

Hinweis:
Kommentieren Sie die wichtigsten Schritte jeder Methode. Achten Sie auf die Namens- und Formatierungskonventionen. Beachten Sie einen einheitlichen Programmierstil. (Bsp.: Nutzen Sie deutsche oder englische Begriffe, aber nicht gemischt. Bitte geben Sie mit @author in allen Klassen einen Autor/ eine Autorin an, bzw. beide AuthorenInnen.

Ihre Mitfahrdienst-Bibliothek zum ersten Prototyp hat folgende Klasse:

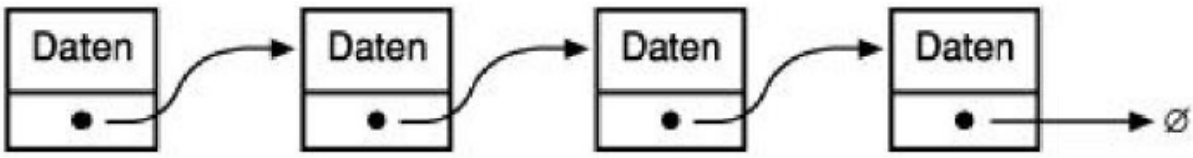
Fahrten-Service (4 Punkte) (Java- Dateien: 1 Interface + 1 Service-Klasse, die das Interface implementiert und die Farten verwaltet)

Das Programm benötigt eine geeignete Schnittstelle für die Fahrtenzugriffe, über diese

- Fahrten (mit Teilfahrten und mit freie Sitzplatzanzahl) eingesehen (übersichtliche Konsolen-Ausgabe) werden können,
- sowie Fahrten gebucht
- und Fahrten storniert werden können,
- und auch Buchungen bestätigt,
- und Buchungen abgelehnt werden können.

Fahrten bzw. Teilfahrten (4 Punkte) (Java- Dateien: 1)

Fahrten können aus mehreren Teilfahrten bestehen. Eine Reise von Berlin nach Köln beispielsweise hat Zwischenstopps bei Magdeburg, Hannover und Bielefeld.
Lösen Sie diese Struktur mit verketteten Listen (einfach verkettet).



Jede Fahrt ist ein Transportmittel zugeordnet.

Transportmittel (3 Punkte) (Java- Dateien: 1 Interface + 1 abstrakte Klasse + 2 weitere Klassen)

Transportmittel werden als abstrakte Klasse definiert. Es werden davon Klassen für private PKW oder lizensierte Reisebusse abgeleitet. Die Transportmittel können Personen aufnehmen. (Siehe folgende Aufgabe)

a. Für einen einheitlichen Zugriff auf die Transportmittel, stellen Sie bitte ein Transportmittel-Interface bereit. Benötigte Methoden und/oder Parameter in dem Interface sind für die Fahrzeugsitzplätze zu entwerfen.

Hinweis:
Es starten alle Personen am Startpunkt und es werden nicht unterwegs noch Fahrgäste aufgenommen. Wer diese Fälle gerne berücksichtigen will, kann dies gerne umsetzen.

Beifahrer mit maximaler Sitzplatzanzahl im Transportmittel (3 Punkte) (1 weitere Klasse)

Lösen Sie das Problem, dass die Fahrzeuge der Fahrten nur eine maximale Sitzplatzanzahl haben und demnach nicht überbucht werden darf. Geben Sie eine Rückmeldung (Buchung abgelehnt), wenn dies der Fall ist.
Implementieren Sie eine Klasse Person. Bauen Sie nötige Parameter wie Name usw. ein und fügen Sie entsprechende Methoden bzw. Getter/Setter ein.

Fahrkosten (2 Punkte) (keine weitere Klasse nötig)

Berechnen Sie die Fahrkosten, indem Sie die verkette Liste um einen Parameter erweitern.

Anwendung (4 Punkte) (1 "JUnit"-Klasse)

Testen Sie das Programm, in dem Sie für die implementierten Methoden der Fahrten ausreichend viele Unit Tests mithilfe von JUnit implementieren. Kommentieren Sie die Tests ausreichend.

a. Legen Sie mind. 2 verschiedene Transportmittel und mind. 4 Beispielfahrten an. (Stichwort JUnit Before)
b. Buchen und stornieren Sie weitere Fahrten und lassen Sie sich entsprechende Rückmeldung auf der Konsole über die Buchungsbestätigung inkl. Fahrzeugkosten ausgeben. Nutzen Sie nur die Methoden der Schnittstelle des Fahrten-Service. Prüfen Sie entsprechend mit JUnit, ob das gewünschte Resultat umgesetzt wurde. Kommentieren Sie alle JUnit-Tests, was genau getestet wird.
c. Buchen Sie mehrere Personen zu einer Fahrt und lösen Sie eine Überbuchung einer Fahrt/ eines Transportmittels aus. Lassen Sie bitte entsprechende Rückmeldung auf der Konsole melden (Schnittstellenmethode: Buchung abgelehnt) und prüfen Sie mit JUnit entsprechend.

Implementieren Sie die Klassen möglichst effizient, d.h. so, dass Sie unter Berücksichtigung der Vererbung unnötige Wiederholungen im Code vermeiden. Kommentieren Sie das Programm ausreichend.

Laden Sie alle Klassen in einem ZIP-File hoch. Das ZIP-File benennen Sie bitte mindestens mit Ihrem/Ihren Nachnamen! (sonst Punktabzug)

(Gesamt 20 Punkte)