
Klassendiagramm CarSharing

Name:

Abgabezeit:

Einleitung

Für einen neuen Anbieter am car sharing Markt soll eine Software zur Verwaltung erstellt werden. In dem System werden sowohl die Stammdaten der Kunden als auch die verfügbaren Fahrzeuge verwaltet. Für Fahrzeuge sollen Modell, Marke und Kennzeichen erfasst werden.

Ein Leihvorgang wird intern als 'Trip' bezeichnet. Der Lebenszyklus eines Trips beginnt mit der Erstellung, etwa durch Reservierung eines bestimmten Fahrzeugs durch einen bestimmten Kunden und einem bestimmten Tarif. Danach wird ein Trip an einem bestimmten Ort und zu einem bestimmten Zeitpunkt gestartet und schlussendlich an einem bestimmten Ort und zu einer bestimmten Zeit beendet. All diese Daten werden für jeden Trip erfasst. Das Tarifmodell soll flexibel und vor allem erweiterbar gestaltet sein. Zur Zeit gibt es einen Minutentarif, bei dem die Gebühr von der Dauer des Trips abhängt und einen Langzeittarif, der einen gewissen Sockelbetrag für eine vereinbarte Leihdauer sowie eine zusätzliche Gebühr pro Tag Überschreitung vorsieht. Die aktive Marketingabteilung betont, dass sie ständig neue Tarife entwickelt und hier größtmögliche Flexibilität fordert, das System muss also leicht um neue Tarife ergänzt werden können.

Die Applikation soll Trips schlussendlich filtern, sortieren und darstellen sowie als Datei speichern können.

Aufgabe

Erstellen Sie ein Klassendiagramm mit

- Allen teilnehmenden Klassen, verwenden Sie abstrakte Klassen und Interfaces wo es angebracht ist (5 Punkte)
- Aus der Angabe identifizierten Attributen; alle Attribute dürfen public sein und es müssen keine getter/setter Methoden angegeben werden (4 Punkte)
- Operationen die nicht schon durch die Attribute dargestellt werden (daher Getter/Setter müssen nicht abgebildet werden) (5 Punkte)
- Assoziationen zwischen den Klassen, geben Sie korrekte Vererbungs-/Realisierungs-Beziehungen an und versuche Sie sinnvoll zwischen Assoziation/Aggregation/Komposition zu wählen (4 Punkte)
- Geben Sie für alle Assoziation/Aggregation/Komposition Multiplizitäten an (2 Punkte)

Punkte

Aufgabe	Punkte
Klassen	5
Attribute	4
Operationen	5
Assoziationen	4
Multiplizitäten	2
Total	20