

C23 – Laborübung 2

Kontext

Ein Taxiunternehmen benötigt ein Programm zur Erfassung und Abrechnung von Fahrten. Das Programm soll für jedes der Taxis den Tageskilometerstand, den aktuellen Tankinhalt und die Geldbilanz erfassen.

Das Unternehmen hat 2 Taxis. Diese unterscheiden sich in Bezug auf maximalen Tankinhalt und Verbrauch, wie auch im verlangten Fahrpreis pro Kilometer:

	Maximaler Tankinhalt	Verbrauch	Fahrpreis
Taxi 1	75 Liter	7,2 Liter / 100 km	0,70 Euro / km
Taxi 2	90 Liter	12.5 Liter / 100 km	0,95 Euro / km

Wenn eine Fahrt ausgeführt wird, müssen sich Tankinhalt und Tageskilometerstand entsprechend ändern. Wurde die Fahrt mit Passagieren durchgeführt, dann bezahlen diese die Fahrt und die Geldbilanz muss sich entsprechend ändern.

Die Fahrzeuge müssen von Zeit zu Zeit aufgetankt werden. Dies verändert den aktuellen Tankinhalt und die Geldbilanz. Der Benzinpreis schwankt zwischen 1,30 und 1,65 Euro pro Liter.

Am Morgen (beim Start des Abrechnungsprogramms) wird angenommen, dass der Tank voll ist, und Tageskilometerstand und Geldbörse den Wert 0 aufweisen.

Aufgabenstellung

Aufgabe 1 – Klassen und Instanzen

- Erstellen Sie eine Klasse „Taxi“ mit den unten aufgeführten Member-Variablen und Member-Funktionen. Erstellen Sie zwei Instanzen dieser Klasse (für die beiden Taxis des Unternehmens).

Member-Variablen

- Tageskilometerstand
- Aktueller Tankinhalt
- Maximaler Tankinhalt
- Verbrauch pro Kilometer
- Fahrpreis pro Kilometer
- Geldbilanz

Member-Funktionen

- initialisieren(...)
Input: Verbrauch, Fahrpreis, maximaler Tankinhalt
 - Zuweisung der Input-Werte an die Member-Variablen
 - Zuweisung der Standard-Werte an die Member-Variablen (Tageskilometerstand = 0, aktueller Tankinhalt = maximaler Tankinhalt, Geldbilanz = 0)
- fahrtVerbuchen(...)
Input: Entfernung, wurden Fahrgäste transportiert?
 - Prüfung ob der vorhandene Tankinhalt für die Fahrt ausreicht, sonst Fehlermeldung
 - Neuberechnung Tageskilometerstand und aktueller Tankinhalt

- Neuberechnung Geldbilanz unter Berücksichtigung der Frage, ob Passagiere transportiert wurden
- tanken(...)
 - Input: aktueller Benzinpreis
 - Auffüllen des Tanks bis zum maximalen Tankinhalt (sofern die Geldbilanz dies erlaubt), oder soweit das vorhandene Geld reicht
 - Neuberechnung Geldbilanz
- print()
 - Ausgabe der aktuellen Werte für Tageskilometerstands, Tankinhalt und Geldbilanz

Hinweis: Die Input-Informationen sollen jeweils als Funktionsparameter an die Funktionen übergeben werden. In den Member-Funktionen der Klasse sollen keine Nutzereingabe erfolgen!

Aufgabe 2 – Hauptprogramm

- a. Schreiben Sie ein Hauptprogramm, das jeweils zwei Instanzen der Klasse „Taxi“ erstellt und testet:
 - Fahrt verbuchen (mit Fahrgast)
 - Fahrt verbuchen (ohne Fahrgast)
 - Tanken
 - Ausgeben

Aufgabe 3 – Programmstruktur mit Nutzereingaben

- a. Ergänzen Sie das Programm um ein Benutzermenü:
 - Die 4 Funktionen (Fahrt verbuchen mit oder ohne Fahrgast, Tanken, Ausgeben) sollen über ein Nutzermenü erreichbar sein.
 - Der Nutzer soll wählen können, für welches der beiden Taxis verbucht / getankt / ausgegeben werden soll.
 - Das Nutzermenü soll auch das Beenden des Programms gestatten.
- b. Stellen Sie sicher, dass in den Teilmenüs fehlerhafte Eingaben abgefangen werden. Reagieren Sie auf fehlerhafte Eingaben mit einer Fehlermeldung an den Benutzer und lassen Sie ihn die Eingabe wiederholen.

Aufgabe 4 – Konstruktoren und Qualifier

- a. Ersetzen Sie die Initialisierungsfunktion durch einen Konstruktor, der Verbrauch, Fahrpreis und maximalen Tankinhalt als Parameter akzeptiert. Achten Sie darauf, alle Member-Variablen geeignet zu initialisieren.
Löschen Sie die Initialisierungsfunktion nicht aus dem Quelltext, sondern kommentieren Sie sie bitte nur aus.
- b. Wandeln Sie die Member-Variablen für maximalen Tankinhalt, Verbrauch pro Kilometer und Preis pro Kilometer in konstante Member-Variablen um.
Passen Sie, falls nötig, den Konstruktor entsprechend an (Stichwort: Initialisierung der konstanten Member-Variablen).
- c. Wandeln Sie alle Member-Funktionen, bei denen dies sinnvoll ist, in konstante Member-Funktionen um.