

## Praktikum 4 zu Objektorientierte Programmierung

### Aufgabe 4.1 (Aufwand ca. 8 Stunden, Abgabe bis 04.05.2014 um 18 Uhr)

In dieser Aufgabe sollen einige Funktionen einer CarSharing-Anwendung realisiert werden. CarSharing-Unternehmen unterhalten Standorte, an denen Fahrzeuge stationiert sind. Jedes Fahrzeug besitzt einen Namen und einen festen Standort.

Alle Funktionen sollen in einer Klasse `Fahrzeugmanager` realisiert werden, die im Paket `carsharing` liegt. Ein `Fahrzeugmanager` verwaltet die Fahrzeuge eines CarSharing-Unternehmens.

Realisieren Sie in der Klasse `Fahrzeugmanager` folgenden Methoden:

- Einen Konstruktor `Fahrzeugmanager()`, durch den ein Objekt dieser Klasse erzeugt wird. Ein gerade erzeugter `Fahrzeugmanager` verwaltet anfangs keine Fahrzeuge.
- Eine Instanzmethode `void fuegeFahrzeugHinzu(String, String)`, durch die dem `Fahrzeugmanager` ein Fahrzeug mit einem bestimmten Namen (erster Parameter) und Standort (zweiter Parameter) hinzugefügt wird. Falls der `Fahrzeugmanager` bereits ein Fahrzeug mit diesem Namen verwaltet, soll *kein* Fahrzeug hinzugefügt werden.
- Eine Instanzmethode `ArrayList<String> gibFahrzeugnamen()`, die die Namen aller Fahrzeuge *alphabetisch sortiert* zurückgibt.
- Eine Instanzmethode `ArrayList<String> gibFahrzeugnamen(String)`, die *alphabetisch sortiert* die Namen aller Fahrzeuge des angegebenen Standorts zurückgibt.
- Eine Instanzmethode `boolean bucheFahrzeug(String, String, String)`, mit der das Fahrzeug mit dem angegebenen Namen (erster Parameter) für einen bestimmten Zeitraum gebucht wird. Der Beginn des Zeitraums wird durch den zweiten Parameter, das Ende durch den dritten Parameter angegeben. Die Angabe der Zeitpunkte erfolgt im Format `JJJJ/MM/TT HH:MM`, z. B. `2005/04/22 09:35` für den 22. April 2005, 9:35 Uhr.

Die Methode liefert genau dann `true` zurück, wenn das angegebene Fahrzeug für den gewünschten Zeitraum gebucht werden kann, d. h. sich der gewünschte Buchungszeitraum mit keiner anderen Buchung dieses Fahrzeugs überschneidet. Kurz gesagt: das Fahrzeug ist in der gewünschten Zeit frei.

- Eine Instanzmethode `ArrayList<String> gibVerfuegbareFahrzeuge(String, String, String)`, die *alphabetisch sortiert* die Namen aller Fahrzeuge des angegebenen

Standorts (erster Parameter) zurückgibt, die in einem bestimmten Zeitraum (zweiter und dritter Parameter) verfügbar sind. Ein Fahrzeug ist in einem Zeitraum verfügbar, wenn es für diesen Zeitraum gebucht werden kann.

Sie können davon ausgehen, dass alle Methoden mit *gültigen* Parametern aufgerufen werden, d. h. bei Aufruf der Methoden werden nur

- gültige Namen für Fahrzeuge und Standorte übergeben,
- das Format der Zeit- und Datumsangaben ist korrekt,
- der Beginn eines Zeit- oder Datumsbereichs liegt nicht später als dessen Ende.

Realisieren Sie außerdem eine Testklasse `FahrzeugmanagerTest` basierend auf `JUnit`. Die Klasse soll ebenfalls im Paket `carsharing` liegen (im NetBeans-Projekt gibt es also unter `src` *und* `test` ein Verzeichnis `carsharing`). Erzeugen Sie als Testobjekt einen `Fahrzeugmanager` mit diesen Fahrzeugen

Fahrzeugname	Standort
Rathaus 1	Rathaus
Bahnhof 1	Bahnhof
Bahnhof 2	Bahnhof
Bahnhof 3	Bahnhof

und diesen Buchungen:

Fahrzeugname	Buchung von	bis
Bahnhof 1	14. April 2005, 20:00 Uhr	15. April 2005, 08:00 Uhr
Bahnhof 1	15. April 2005, 18:00 Uhr	16. April 2005, 00:00 Uhr
Bahnhof 2	14. April 2005, 11:00 Uhr	15. April 2005, 12:00 Uhr
Bahnhof 3	15. April 2005, 10:00 Uhr	15. April 2005, 19:00 Uhr

Realisieren Sie auf der Grundlage des Testobjekts *mindestens* folgende Testabläufe:

Test für `gibFahrzeugnamen()`: Sollergebnis: Liste mit Bahnhof 1, Bahnhof 2, Bahnhof 3, Rathaus  
1

Test für gibFahrzeugnamen(String):

Testmuster	Sollergebnis
Standort Bahnhof	Liste mit Bahnhof 1, Bahnhof 2, Bahnhof 3
Standort Rathaus	Liste mit Rathaus 1

Test für gibVerfuegbareFahrzeuge:

Testmuster	Sollergebnis
Standort Bahnhof, Zeitraum 15. April 2005, 11:30 Uhr bis 15. April 2005 19:00 Uhr	leere Liste
Standort Bahnhof, Zeitraum 15. April 2005, 12:00 Uhr bis 15. April 2005, 18:00 Uhr	Liste mit Bahnhof 1, Bahnhof 2
Standort Bahnhof, Zeitraum 15. April 2005, 19:15 Uhr bis 15. April 2005, 23:00 Uhr	Liste mit Bahnhof 2, Bahnhof 3

Test für bucheFahrzeug:

Testmuster	Sollergebnis
Fahrzeug Bahnhof 3, 15. April 2005, 09:00 Uhr bis 15. April 2005, 10:00 Uhr	true
Fahrzeug Bahnhof 3, 15. April 2005, 09:00 Uhr bis 15. April 2005, 11:00 Uhr	false
Fahrzeug Bahnhof 3, 15. April 2005, 11:00 Uhr bis 15. April 2005, 18:00 Uhr	false
Fahrzeug Bahnhof 3, 15. April 2005, 18:00 Uhr bis 15. April 2005, 20:00 Uhr	false
Fahrzeug Bahnhof 3, 15. April 2005, 19:00 Uhr bis 15. April 2005, 20:00 Uhr	true
Fahrzeug Bahnhof 3, 15. April 2005, 9:00 Uhr bis 15. April 2005, 20:00 Uhr	false

# Hinweise

- Achten Sie darauf, dass in den Eigenschaften Ihres NetBeans-Projekts die Zeichenkodierung UTF-8 eingestellt ist. Ist dies nicht der Fall, kann es bei der automatischen Auswertung Ihrer Lösung zu Fehlern beim Compilieren kommen. Die Lösung wird dann mit einer Erfolgsquote von 0% gewertet.
- Verwenden Sie nur den Vorlesungsstoff bis einschließlich Kapitel 7.
- Überlegen Sie, welche Objekte mit welchen Methoden sinnvoll sind oder benötigt werden. Dies ist ein planerischer, gedanklicher Prozess, für den Sie Papier und Bleistift benötigen, aber keinen Rechner, und für den sich Teamarbeit anbietet.
- Sie dürfen selbstverständlich zusätzliche Klassen und Methoden realisieren. Zusätzliche Klassen müssen ebenfalls im Paket `carsharing` liegen.
- Es geht in dieser Aufgabe insbesondere darum zu üben, welche Collection-Klasse für welchen Zweck am besten eingesetzt werden kann. Machen Sie sich also darüber Gedanken und sagen Sie sich nicht „... ich nehme *immer* Arrays oder Listen, mit denen kenne ich mich am besten aus ...“.
- Die Zeitangaben der Methoden `bucheFahrzeug` und `gibVerfuegbareFahrzeuge` bezeichnen *Zeitpunkte*. Es ist deshalb z. B. möglich, für ein Fahrzeug eine Buchung bis 10:00 Uhr und eine andere Buchung ab 10:00 Uhr zu haben.
- Schauen Sie sich die Dokumentation der Klasse `java.util.Collections` an. Vielleicht gibt es in dieser Klasse eine Methode, die Ihnen beim Sortieren hilft.
- Denken Sie an die ausreichende Dokumentation und Kommentierung Ihrer Lösung. Beachten Sie die unterschiedliche Bedeutung der *externen Dokumentation* `/** ... */` vor einer Klasse oder Methode und des *Implementierungskommentars* `/* ... */` innerhalb einer Methode. Die externe Dokumentation sagt, *was* eine Klasse oder Methode leistet, der Implementierungskommentar hilft zu verstehen, *wie* es gemacht wird. Verwenden Sie Implementierungskommentare vor allem, um den Berechnungsablauf verständlich zu machen.
- Erzeugen Sie die HTML-Dokumentation Ihrer Klasse und überzeugen Sie sich, ob Ihre externe Dokumentation sinnvoll und ohne Kenntnis des Quellcodes der Klasse hilfreich ist.
- Erstellen Sie je ein zip-Archiv des Quellordners `carsharing` unter `src` und des gleichnamigen Ordners unter `test` und laden Sie beide zu Moodle hoch.
- Im Veranstaltungskalender finden Sie die Termine, an denen diese Lösung im Praktikum besprochen wird. Bringen Sie zu diesen Terminen bitte die Auswertung Ihrer Lösung mit, entweder ausgedruckt oder unmittelbar auf Ihrem Rechner verfügbar.