Prof. Dr. Ingo Claßen
Prof. Dr. Martin Kempa
Patrick Dohmeier

3. Übungsblatt: JDBC - Anwendungslogik in Java

Für die folgende Aufgabe muss das Datenbankschema der Miniwelt *Mauterhebung* in Ihrer Datenbank vorhanden sein.

Sämtliche Vorgaben befinden sich in der Datei ***maut_service.zip***, die ein vorbereitetes Eclipse-Projekt enthält und das Sie von der Webseite dieser Lehrveranstaltung herunterladen müssen. Dieses Projekt benötigt einige Java-Bibliotheken, die sich ebenfalls als Eclipse-Projekt in der Datei dbtech_lib.zip befinden.

Beide Projekte können Sie über den Menüpunkt **File > Import > Existing Projects into Workspace** (Zip-Datei unter Punkt select archive file: auswählen, Finish aktivieren) Ihrem Eclipse-Arbeitsbereich hinzufügen. Das Projekt ***maut_service*** enthält bereits Klassen des Anwendungssystems, wie die zu verwendenden Ausnahmen im Paket de.htwberlin.exceptions. Diese Klassen dürfen Sie nur dann verändern, wenn es in der Aufgabe angegeben ist.

Während der Entwicklung des Programmcodes können Sie jederzeit Ihren aktuellen Zustand testen. Dafür gibt es eine Testklasse mit Namen ***MautServiceTest.java*** (im Paket ***de.htwberlin.maut.test***). Die Testklasse basiert auf den Testframeworks JUnit und DBUnit, mit denen automatisierte Tests unterstützt werden. In die Eclipse Entwicklungsumgebung ist eine Testausführungskomponente integriert, mit der Sie die Tests durchführen können. Dazu aktivieren Sie aus dem Kontextmenü der Testklasse, das Sie über die rechte Maustaste erreichen, den Menüpunkt Run as > JUnit Test.

Für das Test- und Anwendungssystem ist eine Datenbankverbindung notwendig. Diese konfigurieren Sie über die Datei ***DbCred.java*** (im Paket ***de.htwberlin.utils***). Dort müssen Sie die Werte der Schlüssel dbunit.username, dbunit.password und dbunit.schema an Ihre Bedürfnisse anpassen.

Bedenken Sie bitte, dass eine erfolgreiche Ausführung aller Tests nicht automatisch die Korrektheit Ihrer Lösung sicherstellt. Tests können immer nur die Anwesenheit von Fehlern zeigen, nicht aber deren Abwesenheit. Das liegt daran, dass Tests im Allgemeinen nicht vollständig alle Fehlersituationen abdecken. In der Bewertung Ihrer Lösung ist daher der erfolgreiche Durchlauf aller Tests eine notwendige Bedingung zum Erreichen der vollen Punktzahl. Es kann aber trotzdem Punktabzug geben, falls Ihre Lösung Fehler enthält, die durch die Tests nicht aufgedeckt werden.

1. Aufgabe (10 Punkte)

In diesem Übungsblatt sollen Sie einen Service mit Anwendungslogik implementieren, der mittels JDBC auf ein Datenbanksystem zugreift. Dabei handelt es sich um die Methode ***berechneMaut***, die die folgende Schnittstelle zeigt.

Prof. Dr. Ingo Claßen
Prof. Dr. Martin Kempa
Patrick Dohmeier

```
/***
       * Die Methode realisiert einen Algorithmus, der die übermittelten
       * Fahrzeugdaten mit der Datenbank auf Richtigkeit überprüft und für einen
       * mautpflichtigen Streckenabschnitt die zu zahlende Maut für ein Fahrzeug
       * im Automatischen Verfahren berechnet.
       *
       * Zuvor wird überprüft, ob das Fahrzeug registriert ist und über ein
       * eingebautes Fahrzeuggerät verfügt und die übermittelten Daten des
       * Kontrollsystems korrekt sind. Bei Fahrzeugen im Manuellen Verfahren wird
       * darüber hinaus geprüft, ob es noch offene Buchungen für den Mautabschnitt
       * gibt oder eine Doppelbefahrung aufgetreten ist. Besteht noch eine offene
       * Buchung für den Mautabschnitt, so wird diese Buchung für das Fahrzeug auf
       * abgeschlossen gesetzt.
       *
       * Sind die Daten des Fahrzeugs im Automatischen Verfahren korrekt, wird
       * anhand der Mautkategorie (die sich aus der Achszahl und der
       * Schadstoffklasse des Fahrzeugs zusammensetzt) und der Mautabschnittslänge
       * die zu zahlende Maut berechnet, in der Mauterhebung gespeichert und
       * letztendlich zurückgegeben.
       *
       *
       * @param mautAbschnitt
       *            - identifiziert einen mautpflichtigen Abschnitt
       * @param achszahl
       *            - identifiziert die Anzahl der Achsen für das Fahrzeug das
       *            durch ein Kontrollsystem erfasst worden ist
       * @param kennzeichen
       *            - idenfiziert das amtliche Kennzeichen des Fahrzeugs das durch
       *            das Kontrollsystem erfasst worden ist
       * @throws UnkownVehicleException
       *             - falls das Fahrzeug weder registriert ist, noch eine offene
       *             Buchung vorliegt
       * @throws InvalidVehicleDataException
       *             - falls Daten des Kontrollsystems nicht mit den hinterlegten
       *             Daten in der Datenbank übereinstimmt
       * @throws AlreadyCruisedException
       *             - falls eine Doppelbefahrung für Fahrzeuge im Manuellen
       *             Verfahren vorliegt
       * @return die berechnete Maut für das Fahrzeug im Automatischen Verfahren
       *         auf dem Streckenabschnitt anhand der Fahrzeugdaten
       */
      float berechneMaut(int mautAbschnitt, int achszahl, String kennzeichen)
                  throws UnkownVehicleException, InvalidVehicleDataException,
                  AlreadyCruisedException;

   }
```

Ihre Aufgabe besteht darin, die Klasse ***MautServiceImpl.java*** zu realisieren, deren Funktionalität als JavaDoc-Kommentare in der Schnittstelle ***IMauterhebung.java*** beschrieben wird. Dabei dürfen Sie sämtliche Klassen der Vorgabe verwenden, aber nicht verändern. Es empfiehlt sich, mehrere Methoden zu entwickeln, die einander aufrufen. In dem Fall sind alle Methoden in der Klasse MauterhebungImpl zu verwalten. Die in der Beschreibung genannten Ausnahmeklassen (Exceptions) liegen im Projekt bereits vor und müssen bei der Lösung verwendet werden.

Prof. Dr. Ingo Claßen
Prof. Dr. Martin Kempa
Patrick Dohmeier

In der Aufgabestellung übernehmen Sie die Position einer Kontrollbrücke, die die Daten auf Richtigkeit prüft und für unsere Übungszwecke erhebt, bei Bedarf berechnet und speichert.

Ein Fahrzeug ist im Automatischen Verfahren, wenn es ein Fahrzeuggerät verbaut hat. Ist dies der Fall, muss die Höhe der Maut anhand der spezifischen Fahrzeugdaten berechnet und gespeichert werden. Ist kein Fahrzeuggerät verbaut, befindet sich das Fahrzeug im Manuellen Verfahren. Auch hier werden die spezifischen Fahrzeugdaten auf Richtigkeit überprüft und die offene Buchung auf geschlossen gesetzt. Im Manuellen Verfahren muss die Berechnung der Maut nicht durchgeführt werden, da die Maut schon vor Fahrtbeginn entrichtet werden muss.

Die Formel zur Berechnung der Maut für einen mautpflichtigen Abschnitt lautet:

Mautsatz je km in € * Mautabschnittslänge in km = Maut in €

Das folgende Beispiel zeigt die Berechnug der Maut für einen mautpflichtigen Abschnitt anhand fahrzeugspezifischer Daten:

Der Mautabschnitt mit der ID 92 hat eine Länge von 5700 m -> 5,7 km
Das Fahrzeug mit dem amtlichen Kennzeichen HH 3754 hat 3 Achsen und die Mautschadstoffklasse S7. Das entspricht in der Mautkategorie (Achszahl = 3 && SSKL_ID = 7) 11,3 Cent je km, also 0,113 € je km für einen mautpflichtigen Kilometer.

In unsere Formel eingesetzt ergibt diese Abschnittsbefahrung für das Fahrzeug mit den spezifischen Daten

0,113 € * 5,7 km = 0,64 €

2. Aufgabe (5 Punkte)

Struktieren Sie Ihren Code so, dass Sie eines der Architektur-Muster Table-Data-Gateway, Row-Data-Gateway oder Data-Access-Object verwenden.

Zusatzaufgabe (2 Punkte)

Präsentieren und erklären Sie Ihre erarbeitete Lösung in der nächsten Übung.

Abgabe

Termin und Modus wird durch den Lehrenden der jeweiligen Übung festgelegt.