

4. Übungsblatt: PL / SQL

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

Sämtliche Vorgaben befinden sich in der Datei **dbtech_procedure.zip**, (Sie benötigen ebenfalls wieder zusätzlich das Projekt **dbtech_lib.zip**) die ein vorbereitetes Eclipse-Projekt enthält und die Sie von der Webseite (Moodle) dieser Lehrveranstaltung herunterladen müssen. Es enthält Klassen und Ausnahmeklassen, die den Aufruf der Stored-Procedure realisieren. Eine Änderung dieser Klassen ist nicht notwendig.

Zum Testen Ihres Codes gibt es eine Testklasse mit Namen **MautServiceTest.java** (im Paket **de.htwberlin.maut.test**), die Ihre Prozedur in der Datenbank aufruft. Wie in der vorherigen Aufgabe muss die Datenbankverbindung für das Testsystem über die Datei **DbCred.java** (im Paket **de.htwberlin.utils**) konfiguriert werden.

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

(15 Punkte)

In dieser Aufgabe sollen Sie einen Service mit Anwendungslogik in PL/SQL implementieren. Dabei handelt es sich um die Prozedur **BERECHNEMAUT**, die die folgende Paket-Schnittstelle zeigt:

create or replace

PACKAGE maut_service **AS**

-- Exception wird ausgelöst, falls das Fahrzeug nicht bekannt ist.
-- D.h. es ist nicht im Automatischen Verfahren unterwegs und es liegt keine
-- offene Buchung vor

UNKOWN_VEHICLE EXCEPTION;

PRAGMA EXCEPTION_INIT(UNKOWN_VEHICLE, -20001);

-- Exception wird ausgelöst, falls das Fahrzeug mit der falschen
-- Achszahl unterwegs ist.

INVALID_VEHICLE_DATA EXCEPTION;

PRAGMA EXCEPTION_INIT(INVALID_VEHICLE_DATA, -20002);

-- Exception wird ausgelöst, falls das Fahrzeug den Mautabschnitt
-- bereits befahren hat und keine offene Buchung vorliegt.

```
ALREADY_CRUISED EXCEPTION;  
PRAGMA EXCEPTION_INIT(ALREADY_CRUISED, -20003);  
/**  
 * 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.  
  
 * Parameter p_mautabschnitt identifiziert den mautpflichtigen Abschnitt  
 * Parameter p_achszahl identifiziert die Anzahl der Achsen des Fahrzeugs  
 * Parameter p_kennzeichen identifiziert das amtliche Kennzeichen des Fahrzeugs  
 */  
PROCEDURE BERECHNEMAUT(  
  P_MAUTABSCHNITT MAUTABSCHNITT.ABSCHNITTS_ID%TYPE,  
  P_ACHSZAHL    FAHRZEUG.ACHSEN%TYPE,  
  P_KENNZEICHEN FAHRZEUG.KENNZEICHEN%TYPE);  
END maut_service;
```

Realisieren Sie den Paket-Body **maut_service** in PL/SQL passend zur angegebenen Paket-Spezifikation. Für den Fall, dass Sie mehrere Prozeduren (oder Funktionen) entwickeln, die einander aufrufen, sind alle Prozeduren (oder Funktionen) im Paket **maut_service** zu verwalten.

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.