

Web Engineering

Java Database Connectivity (JDBC)

Task 1 (JDBC) (20 points)

Define Java classes `Model` and `Vehicle` with the following properties:

`Model.java`

```
modelId:int;  
name:String;  
vehicles:List<Vehicle>;
```

`Vehicle.java`

```
vehicleId:int;  
name:String;  
distance:long;  
enginePower:int;  
productionYear:int;  
price:long;  
model:Model;
```

The classes should implement appropriate `getXxx()` and `setXxx()`, as well as methods `load()`, `insert()`, `delete()` and `update()`, functioning as follows:

- `load()` – for the defined value of unique identifier, i.e., primary key loads all property values,
- `insert()` – inserts a new record into the appropriate table,
- `update()` – updates a table record which corresponds to the instance of the class,
- `delete()` – drops the appropriate record from the appropriate table.

Since the relationship between entities `Model` and `Vehicle` is 1:n the corresponding classes must ensure a proper handling of the relationship, i.e., the classes should implement the methods to add, delete and load instance(s) of the classe(s) from the other part of the relationship (for example,

for the class Model that would be methods addVehicle(Vehicle), removeVehicle(Vehicle) and List<Vehicle> getVehicles(), but for the class Vehicle the methods could be setModel(Model) and Model getModel().

Write a test case which illustrates the usage of ALL implemented methods.

APPENDIX: DB SCRIPT

```
CREATE DATABASE autoshop DEFAULT CHARACTER SET utf8;
USE autoshop;

CREATE TABLE model
(
    modelId          INTEGER          NOT NULL,
    modelName        VARCHAR(50)     NOT NULL,
    PRIMARY KEY (modelId)
);

CREATE TABLE vehicle
(
    vehicleId        INTEGER          NOT NULL,
    modelId          INTEGER          NOT NULL,
    name             VARCHAR(150)    NOT NULL,
    distance         BIGINT          NOT NULL,
    enginePower      INTEGER          NOT NULL,
    productionYear   INTEGER          NOT NULL,
    price           BIGINT          NOT NULL,
    PRIMARY KEY (vehicleId)
);

ALTER TABLE vehicle ADD CONSTRAINT fk_model FOREIGN KEY (modelId)
    REFERENCES model (modelId) ON DELETE CASCADE ON UPDATE CASCADE;
```