**ЗВІТ**

до лабораторної роботи №2

з предмету “Методи об’єктно орієнтованого прогрумування”

Виконав

Студент групи БІ-1

Франчук Іван

1. Постановка задачі

Розробити програму, що забезпечує введення і редагування інформації про об'єкти відповідно до заданої предметної області. Інформація про об'єкти повинна зберігатися в окремій базі даних.

Доступ до даних здійснюється з використанням засобів JDBC або ODBC.

2. Вихідний код програми

Клас “Модель”:

import java.sql. \*;  
  
public class Model {  
 private final Connection con; // connection to db  
 private final Statement stmt; // operator  
  
 // constructor  
 public Model(String DBName, String ip, int port)  
 throws Exception {  
  
 String url = "jdbc:mysql://" + ip + ":" + port + "/" +  
 DBName + "?serverTimezone=Europe/Kiev&useSSL=FALSE";  
 con = DriverManager.*getConnection*(url, "admin", "Password\_1");  
 stmt = con.createStatement();  
 }  
  
 // models list  
 public void showModels() {  
 String sql = "SELECT ID, Name, ManufacturerID, ColorID, Year, EngineCapacity, Count FROM Model";  
 try {  
 ResultSet rs = stmt.executeQuery(sql);  
 System.*out*.println("Cars list:");  
 System.*out*.println("ID - Name - ManufacturerID - ColorID - Year - EngineCapacity - Count");  
 while (rs.next()) {  
 int id = rs.getInt("ID");  
 String name = rs.getString("Name");  
 int man\_id = rs.getInt("ManufacturerID");  
 int col\_id = rs.getInt("ColorID");  
 int year = rs.getInt("Year");  
 int eng\_cap = rs.getInt("EngineCapacity");  
 int count = rs.getInt("Count");  
 System.*out*.println(">>" + id + " - " + name + " - " + man\_id + " - " + col\_id + " - " + year +  
 " - " + eng\_cap + " - " + count);  
 }  
 rs.close();  
 } catch (SQLException e) {  
 System.*out*.println(  
 "ERROR while getting auto's list");  
 System.*out*.println(" >> " + e.getMessage());  
 }  
 }  
  
 // models by manufacturer ID  
 public void findModelsByManID(int manufacturer\_id) {  
 String sql = "SELECT ID, Name, ManufacturerID, ColorID, Year, EngineCapacity, Count FROM Model " +  
 "WHERE ManufacturerID = " + manufacturer\_id;  
 try {  
 ResultSet rs = stmt.executeQuery(sql);  
 System.*out*.println("Cars list:");  
 System.*out*.println("ID - Name - ManufacturerID - ColorID - Year - EngineCapacity - Count");  
 while (rs.next()) {  
 int id = rs.getInt("ID");  
 String name = rs.getString("Name");  
 int man\_id = rs.getInt("ManufacturerID");  
 int col\_id = rs.getInt("ColorID");  
 int year = rs.getInt("Year");  
 int eng\_cap = rs.getInt("EngineCapacity");  
 int count = rs.getInt("Count");  
 System.*out*.println(">>" + id + " - " + name + " - " + man\_id + " - " + col\_id + " - " + year +  
 " - " + eng\_cap + " - " + count);  
 }  
 rs.close();  
 } catch (SQLException e) {  
 System.*out*.println(  
 "ERROR while getting auto's list");  
 System.*out*.println(" >> " + e.getMessage());  
 }  
 }  
  
 // stop work  
 public void stop() throws SQLException {  
 con.close();  
 }  
  
 // add model  
 public boolean addModel(String name, int man\_id, int col\_id, int year, int eng\_cap, int count) {  
 String sql = "INSERT INTO Model (Name, ManufacturerID, ColorID, Year, EngineCapacity, Count) " +  
 "VALUES ('" + name + "', " + man\_id + ", " + col\_id + ", "  
 + year + ", " + eng\_cap + ", " + count + ")";  
 try {  
 stmt.executeUpdate(sql);  
 System.*out*.println("Model " + name + " added successfully");  
 return true;  
 } catch (SQLException e) {  
 System.*out*.println("ERROR! Model " + name + " not added!");  
 System.*out*.println(" >> " + e.getMessage());  
 return false;  
 }  
 }  
  
 // update model  
 public boolean updateModel(int id, String name, int man\_id, int col\_id, int year, int eng\_cap, int count) {  
 String sql = "UPDATE Model SET ID = " + id;  
 if (!name.equals("")) {  
 sql += ", Name = '" + name + "'";  
 }  
  
 if (man\_id != 0) {  
 sql += ", ManufacturerID = " + man\_id;  
 }  
  
 if (col\_id != 0) {  
 sql += ", ColorID = " + col\_id;  
 }  
  
 if (year > 0) {  
 sql += ", Year = " + year;  
 }  
  
 if (eng\_cap > 0) {  
 sql += ", EngineCapacity = " + eng\_cap;  
 }  
  
 if (count >= 0) {  
 sql += ", Count = " + count;  
 }  
 sql += " WHERE ID = " + id;  
  
 try {  
 stmt.executeUpdate(sql);  
 System.*out*.println("Model " + name + " added successfully");  
 return true;  
 } catch (SQLException e) {  
 System.*out*.println("ERROR! Model " + name + " not added!");  
 System.*out*.println(" >> " + e.getMessage());  
 return false;  
 }  
 }  
  
 // delete model  
 public boolean deleteModel(int id) {  
 String sql = "DELETE FROM Model WHERE ID =" + id;  
 try {  
 int c = stmt.executeUpdate(sql);  
 if (c > 0) {  
 System.*out*.println("Model with id " + id + " deleted successfully!");  
 return true;  
 } else {  
 System.*out*.println("Model with id " + id + " not found!");  
  
 return false;  
 }  
 } catch (SQLException e) {  
 System.*out*.println("ERROR while deleting model with id " + id);  
 System.*out*.println(" >> " + e.getMessage());  
 return false;  
 }  
 }  
  
 public static void main(String[] args) throws Exception {  
 }  
}

Клас “Виробник”:

import java.sql. \*;  
import java.util.Calendar;  
  
public class Manufacturer {  
 private final Connection con; // connection to db  
 private final Statement stmt; // operator  
  
 // constructor  
 public Manufacturer(String DBName, String ip, int port)  
 throws Exception {  
  
 String url = "jdbc:mysql://" + ip + ":" + port + "/" +  
 DBName + "?serverTimezone=Europe/Kiev&useSSL=FALSE&allowPublicKeyRetrieval=true";  
 con = DriverManager.*getConnection*(url, "admin", "Password\_1");  
 stmt = con.createStatement();  
 }  
  
 // manufacturers list  
 public void showManufacturers() {  
 String sql = "SELECT ID, Name, FoundationDate FROM Manufacturer";  
 try {  
 ResultSet rs = stmt.executeQuery(sql);  
 System.*out*.println("Manufacturers list:");  
 while (rs.next()) {  
 int id = rs.getInt("ID");  
 String name = rs.getString("Name");  
 Date foundation = rs.getDate("FoundationDate");  
 System.*out*.println(">>" + id + " - " + name + " - " + foundation.toString());  
 }  
 rs.close();  
 } catch (SQLException e) {  
 System.*out*.println(  
 "ERROR while getting Manufacturer`s list");  
 System.*out*.println(" >> " + e.getMessage());  
 }  
 }  
  
 public String getManufacturerFoundationDate(int id) {  
 String sql = "SELECT FoundationDate FROM Manufacturer WHERE ID = " + id;  
 try {  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next()) {  
 Date foundation = rs.getDate("FoundationDate");  
 return foundation.toString();  
 }  
 rs.close();  
 } catch (SQLException e) {  
 System.*out*.println(  
 "ERROR while getting Manufacturer foundation date");  
 System.*out*.println(" >> " + e.getMessage());  
 }  
  
 return "1970-01-01";  
 }  
  
 // stop work  
 public void stop() throws SQLException {  
 con.close();  
 }  
  
 // add manufacturer  
 public boolean addManufacturer(String name, Calendar foundation\_date) {  
 int month = foundation\_date.get(Calendar.*MONTH*);  
 String month\_q;  
 if (month < 10) {  
 month\_q = "0" + String.*valueOf*(month);  
 } else {  
 month\_q = String.*valueOf*(month);  
 }  
  
 int day = foundation\_date.get(Calendar.*DAY\_OF\_MONTH*);  
 String day\_q;  
 if (day < 10) {  
 day\_q = "0" + String.*valueOf*(day);  
 } else {  
 day\_q = String.*valueOf*(day);  
 }  
  
 String sql = "INSERT INTO Manufacturer (Name, FoundationDate) " +  
 "VALUES ('" + name + "', '" + foundation\_date.get(Calendar.*YEAR*) + '-' +  
 month\_q + '-' + day\_q + "')";  
 try {  
 stmt.executeUpdate(sql);  
 System.*out*.println("Manufacturer " + name + " added successfully");  
 return true;  
 } catch (SQLException e) {  
 System.*out*.println("ERROR! Manufacturer " + name + " not added!");  
 System.*out*.println(" >> " + e.getMessage());  
 return false;  
 }  
 }  
  
 public boolean updateManufacturer(int id, String name, String foundation\_date) {  
 String sql;  
 if (name.equals("")) {  
 sql = "UPDATE Manufacturer SET FoundationDate = '" + foundation\_date + "' WHERE ID = " + id;  
 } else {  
 sql = "UPDATE Manufacturer SET Name = '" + name + "', FoundationDate = '" +  
 foundation\_date + "' WHERE ID = " + id;  
 }  
 try {  
 stmt.executeUpdate(sql);  
 System.*out*.println("Manufacturer " + name + " updated successfully");  
 return true;  
 } catch (SQLException e) {  
 System.*out*.println("ERROR! Manufacturer " + name + " not updated!");  
 System.*out*.println(" >> " + e.getMessage());  
 return false;  
 }  
 }  
  
 // delete Manufacturer  
 public boolean deleteManufacturer(int id) {  
 String sql = "DELETE FROM Manufacturer WHERE ID =" + id;  
 try {  
 int c = stmt.executeUpdate(sql);  
 if (c > 0) {  
 System.*out*.println("Manufacturer with id " + id + " deleted successfully!");  
 return true;  
 } else {  
 System.*out*.println("Manufacturer with id " + id + " not found!");  
  
 return false;  
 }  
 } catch (SQLException e) {  
 System.*out*.println("ERROR while deleting Manufacturer with id " + id);  
 System.*out*.println(" >> " + e.getMessage());  
 return false;  
 }  
 }  
  
 public static void main(String[] args) throws Exception {  
 }  
}

Головний клас, що запускає тестування:

import java.util.Calendar;  
import java.util.GregorianCalendar;  
import java.util.Scanner;  
  
public class JavaToMySql {  
  
 public static void main(String[] args) throws Exception {  
 System.*out*.println("Made by Franchuk I.");  
 System.*out*.println("Group BI-1");  
 System.*out*.println("Lab 2, Variant 5: Car Showroom");  
 System.*out*.println("");  
  
 *runTests*();  
 }  
  
 public static void runTests() throws Exception {  
 System.*out*.println("Start testing manufacturers");  
 *testManufacturers*();  
 System.*out*.println("");  
  
 System.*out*.println("Start testing models");  
 *testModels*();  
 }  
  
 public static void testModels() throws Exception {  
 Model m = new Model("CarShowroom", "localhost", 3306);  
 m.showModels();  
 System.*out*.println("");  
  
 Scanner in = new Scanner(System.*in*);  
  
 // find models by manufacturer id  
 {  
 System.*out*.println("Do You want to find models by manufacturer id (y/n)?");  
 String find = in.next();  
  
 if (find.equals("y")) {  
 System.*out*.println("Do You want to see list of manufacturers (y/n)?");  
 String ans = in.next();  
  
 if (ans.equals("y")) {  
 Manufacturer manufacturer = new Manufacturer("CarShowroom", "localhost", 3306);  
 manufacturer.showManufacturers();  
 System.*out*.println("");  
 }  
  
 System.*out*.println("Enter manufacturer id: ");  
 int man\_id = in.nextInt();  
  
 m.findModelsByManID(man\_id);  
 System.*out*.println("");  
 }  
 }  
  
 // add models  
 {  
 System.*out*.println("Do You want to add models (y/n)?");  
 String add = in.next();  
  
 if (add.equals("y")) {  
 m.addModel("EC8", 14, 4, 2020, 18, 1);  
 m.addModel("i50", 3, 1, 2020, 12, 3);  
  
 m.showModels();  
 System.*out*.println("");  
  
 }  
 }  
  
 // update models  
 {  
 System.*out*.println("Do You want to update models (y/n)?");  
 String s = in.next();  
 if (s.equals("y")) {  
 System.*out*.println("How many models you want to update?");  
 int count\_to\_update = in.nextInt();  
  
 for (int i = 0; i < count\_to\_update; ++i) {  
 System.*out*.println("Enter model id: ");  
 int id = in.nextInt();  
  
 System.*out*.println("Enter model`s new name (0 - for do not change name): ");  
 String name = in.next();  
  
 if (name.equals("0")) {  
 name = new String();  
 }  
  
 int manufacturer\_id = 0, color\_id = 0, year = 0, engine\_capacity = 0, count = 0;  
  
 System.*out*.println("Enter new ManufacturerID (0 - for do not change): ");  
 manufacturer\_id = in.nextInt();  
  
 System.*out*.println("Enter new ColorID (0 - for do not change): ");  
 color\_id = in.nextInt();  
  
 System.*out*.println("Enter new Year (0 - for do not change): ");  
 year = in.nextInt();  
  
 System.*out*.println("Enter new EngineCapacity (0 - for do not change): ");  
 engine\_capacity = in.nextInt();  
  
 System.*out*.println("Enter new Count ('-1' - for do not change): ");  
 count = in.nextInt();  
  
 m.updateModel(id, name, manufacturer\_id, color\_id, year, engine\_capacity, count);  
 }  
  
 m.showModels();  
 System.*out*.println("");  
 }  
 }  
  
 // delete models  
 {  
 System.*out*.println("How many models you want to delete?");  
 int count\_to\_delete = in.nextInt();  
  
 if (count\_to\_delete != 0) {  
 System.*out*.println("Which " + count\_to\_delete + " models you want to delete (id)?");  
 }  
  
 for (int i = 0; i < count\_to\_delete; ++i) {  
 int model\_id = in.nextInt();  
 m.deleteModel(model\_id);  
 }  
 }  
  
 m.showModels();  
 m.stop();  
 }  
  
 public static void testManufacturers() throws Exception {  
 Manufacturer m = new Manufacturer("CarShowroom", "localhost", 3306);  
 m.showManufacturers();  
 System.*out*.println("");  
  
 Scanner in = new Scanner(System.*in*);  
  
 // add manufacturers  
 {  
 System.*out*.println("Do You want to add manufacturers (y/n)?");  
 String add = in.nextLine();  
 if (add.equals("y")) {  
 Calendar calendar = new GregorianCalendar(1920, 1, 30);  
 m.addManufacturer("Mazda", calendar);  
 calendar = new GregorianCalendar(1997, 3, 18);  
 m.addManufacturer("Cherry", calendar);  
  
 m.showManufacturers();  
 }  
 }  
  
 // update manufacturers  
 {  
 System.*out*.println("Do You want to update manufacturers (y/n)?");  
 String s = in.nextLine();  
 if (s.equals("y")) {  
 System.*out*.println("How many manufacturers you want to update?");  
 int count\_to\_update = in.nextInt();  
  
 for (int i = 0; i < count\_to\_update; ++i) {  
 System.*out*.println("Enter manufacturer id: ");  
 int id = in.nextInt();  
  
 System.*out*.println("Enter manufacturer new name (0 - for do not change name): ");  
 String name = in.next();  
  
 if (name.equals("0")) {  
 name = new String();  
 }  
  
 String date;  
 int year, month, day;  
 System.*out*.println("Do You want to set new foundation date (y/n): ");  
 String q = in.next();  
 if (q.equals("y")) {  
 System.*out*.println("Enter new foundation year: ");  
 year = in.nextInt();  
  
 System.*out*.println("Enter new foundation month: ");  
 month = in.nextInt();  
  
 System.*out*.println("Enter new foundation day: ");  
 day = in.nextInt();  
  
 String month\_q;  
 if (month < 10) {  
 month\_q = "0" + String.*valueOf*(month);  
 } else {  
 month\_q = String.*valueOf*(month);  
 }  
  
 String day\_q;  
 if (day < 10) {  
 day\_q = "0" + String.*valueOf*(day);  
 } else {  
 day\_q = String.*valueOf*(day);  
 }  
  
 date = String.*valueOf*(year) + '-' + month\_q + '-' + day\_q;  
 } else {  
 date = m.getManufacturerFoundationDate(id);  
 }  
  
 m.updateManufacturer(id, name, date);  
 }  
  
 m.showManufacturers();  
 }  
 }  
  
 // delete manufacturers  
 {  
 System.*out*.println("How many manufacturers you want to delete?");  
 int count\_to\_delete = in.nextInt();  
  
 if (count\_to\_delete != 0) {  
 System.*out*.println("Which " + count\_to\_delete + " manufacturers you want to delete (id)?");  
 }  
  
 for (int i = 0; i < count\_to\_delete; ++i) {  
 int man\_id = in.nextInt();  
 m.deleteManufacturer(man\_id);  
 }  
  
 m.showManufacturers();  
 }  
  
 m.stop();  
 }  
}

3. Опис програми

У програмі реалізовано 3 класи. Розглянемо кожний з них.

Перший – клас “Модель”. Об’єкт класу “Модель” може змінювати поля у базі даних у відповідній таблиці “Model”. У класі реалізовані операції додавання нового об’єкту, редагування існуючого, видалення існуючого об’єкту та пошук за заданим критерієм (виробником).

Другий клас – “Виробник”. Об’єкт класу “Виробник” може змінювати поля у базі даних у відповідній таблиці “Manufacturer”. У класі реалізовані операції додавання нового об’єкту, редагування існуючого та видалення існуючого об’єкту.

Третій – головний клас для запуску застосунку і тестування. В ньому реалізовані тести для перевірки роботи кожного метода з основних класів. При тестуванні ведеться діалог з користувачем, задля можливості перевірки різних варіантів у ручному режимі.

4. ER-модель бази даних

