МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

**ЛАБОРАТОРНАЯ РАБОТА №3**

«Базовые концепции ЯП Java»

Выполнил

Студент ФЭИС

3- го курса, группы ПО-5

Брич М.Н.

Проверил

Крощенко А. А.

2021

Цель работы: научиться создавать и использовать классы в программах на языке программирования Java.

Вариант №3

Задание № 1:

3) Прямоугольный треугольник, заданный длинами сторон – Предусмотреть возможность определения площади и периметра, а также логический метод, определяющий существует или такой треугольник. Конструктор должен позволить создавать объекты с начальной инициализацией. Реализовать метод equals, выполняющий сравнение объектов данного типа.

Задание № 2:

. Составить программу, которая содержит информацию о наличие автобусов в автобусном парке.

Сведения о каждом автобусе содержат (Bus) содержат:

• Фамилия и инициалы водителя;

• Номер автобуса;

• Номер маршрута;

• Марка;

• Год начала эксплуатации;

• Пробег;

• Местонахождение в настоящий момент времени (парк/маршрут).

Программа должна обеспечивать:

• Формирование данных обо всех автобусах в виде списка;

• Формирование списка автобусов выехавших из парка;

• Формирование списка автобусов оставшихся в парке;

• Список автобусов для заданного номера маршрута;

• Список автобусов, которые эксплуатируются больше 10 лет;

• Список автобусов, пробег у которых больше 100000 км.

• Вывод сведений об автобусах, находящихся на маршруте и об автобусах, оставшихся в

парке.

Код:

**Lab3.java:**import Model.Bus;

import Model.enums.LocationStatus;

import java.sql.Connection;

import java.util.Calendar;

import java.util.List;

public class Lab3 {

public static void main(String[] args) throws Exception {

Task1();

Task2(Integer.parseInt(args[0]));

}

private static void Task1() {

Triangle triangle = new Triangle();

triangle.setTriangle(3f, 4f, 5f);

Triangle triangle2 = new Triangle(3f, 4f, 5f);

Triangle triangle3 = new Triangle();

triangle3.setCathet\_1(5f);

triangle3.setCathet\_2(6f);

triangle3.setHypotenuse(7f);

Triangle[] triangles = new Triangle[]{triangle, triangle2, triangle3};

for (int i = 0; i < triangles.length; i++) {

System.out.println("Треугольник №" + (i + 1) + " :" + triangles[i]);

if (triangles[0].isRectangular()) System.out.println("Треугольник №" + (i + 1) + " - прямоугольный");

else System.out.println("Треугольник №" + (i + 1) + " - непрямоугольный");

System.out.println("Периметр треугольника №" + (i + 1) + " :" + triangles[i].Perimeter());

System.out.format("Площадь треугольника №%d : %.3f \n", i+1 , triangles[i].Square());

}

if (triangle.equals(triangle2)) System.out.println("Треугольники 1 и 2 - равны");

else System.out.println("Треугольники 1 и 2 - не равны");

}

private static void Task2(int routeNumber) {

Connection conn = ManagerDB.connectDB();

// ManagerDB.addElement(conn, new Bus("User name","34664","2","Neman Long",2012,345678L, LocationStatus.ROUTE.toString()));

List<Bus> buses = ManagerDB.selectTable(conn , "bus");

showTable(buses);

showBusInROUTE(buses);

showBusInPARK(buses);

showBusByRouteNumber(buses, routeNumber);

showBusByYear(buses);

showBusByMileage(buses);

}

private static void showTable(List<Bus> buses) {

System.out.println("\n\nФормирование данных обо всех автобусах в виде списка");

for(Bus bus : buses)

bus.print();

}

private static void showBusByYear(List<Bus> buses) {

System.out.println("\n\nСписок автобусов, которые эксплуатируются больше 10 лет");

for(Bus bus : buses)

if(Calendar.getInstance().get(Calendar.YEAR) - bus.getYearOperation() > 10)

bus.print();

}

private static void showBusByMileage(List<Bus> buses) {

System.out.println("\n\nСписок автобусов, пробег у которых больше 100000 км.");

for(Bus bus : buses)

if(bus.getMileage() > 100000)

bus.print();

}

private static void showBusByRouteNumber(List<Bus> buses, int routeNumber) {

System.out.println("\n\nСписок автобусов для заданного номера маршрута (Маршрут №" + routeNumber + ")");

for(Bus bus : buses)

if(Integer.parseInt(bus.getRouteNumber()) == routeNumber)

bus.print();

}

private static void showBusInROUTE(List<Bus> buses) {

System.out.println("\n\nФормирование списка автобусов выехавших из парка");

for(Bus bus : buses) {

if(bus.getLocation().toString().equals(LocationStatus.PARK.toString()))

continue;

bus.print();

}

}

private static void showBusInPARK(List<Bus> buses) {

System.out.println("\n\nФормирование списка автобусов оставшихся в парке");

for(Bus bus : buses) {

if(bus.getLocation().toString().equals(LocationStatus.ROUTE.toString()))

continue;

bus.print();

}

}

}

**ManagerDB.java:**

import Model.Bus;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

public class ManagerDB {

final static String HOST\_DB = "localhost";

final static String USER\_DB = "admin";

final static String PASSWORD\_DB = "root";

final static String NAME\_DB = "SPP";

public static Connection connectDB() {

Connection conn = null;

try{

conn = DriverManager.getConnection("jdbc:mysql://" + HOST\_DB + ":3306/" + NAME\_DB, USER\_DB, PASSWORD\_DB);

}

catch (Exception ex) {

ex.printStackTrace();

}

return conn;

}

public static void addElement(Connection conn, Bus bus) {

String commandSQL = "INSERT INTO `bus`(`fio`, `auto\_number`, `route\_number`, `brand`, `year\_operation`, `mileage`, `location`) VALUES (?,?,?,?,?,?,?)";

try {

PreparedStatement statement = conn.prepareStatement(commandSQL);

statement.setString(1, bus.getFIO());

statement.setString(2, bus.getAutoNumber());

statement.setString(3, bus.getRouteNumber());

statement.setString(4, bus.getBrand());

statement.setInt(5, bus.getYearOperation());

statement.setLong(6, bus.getMileage());

statement.setString(7, bus.getLocation());

statement.execute();

} catch (Exception e) {

e.printStackTrace();

}

}

public static List<Bus> selectTable(Connection conn, String nameTable) {

List<Bus> buses = new ArrayList<>();

try {

Statement st = conn.createStatement();

ResultSet resultSet = st.executeQuery("SELECT \* FROM " + nameTable);

while(resultSet.next()){

buses.add(

new Bus(

resultSet.getString(2),

resultSet.getString(3),

resultSet.getString(6),

resultSet.getString(4),

resultSet.getInt(7),

resultSet.getLong(5),

resultSet.getString(8)));

}

} catch (Exception e) {

e.printStackTrace();

}

return buses;

}

}

**Triangle.java:**

public class Triangle {

Float cathet\_1;

Float cathet\_2;

Float hypotenuse;

public Triangle(Float cathet\_1, Float cathet\_2, Float hypotenuse) {

this.cathet\_1 = cathet\_1;

this.cathet\_2 = cathet\_2;

this.hypotenuse = hypotenuse;

}

public void setTriangle(Float cathet\_1, Float cathet\_2, Float hypotenuse) {

this.cathet\_1 = cathet\_1;

this.cathet\_2 = cathet\_2;

this.hypotenuse = hypotenuse;

}

public Triangle() {

this.cathet\_1 = null;

this.cathet\_2 = null;

this.hypotenuse = null;

}

public double Square()

{

return Math.sqrt(Perimeter()\*(Perimeter()-cathet\_1)\*(Perimeter()-cathet\_2)\*(Perimeter()-hypotenuse));

}

public Float Perimeter()

{

return cathet\_1 + cathet\_2 + hypotenuse;

}

public boolean isRectangular()

{

return Math.pow(cathet\_1, 2) + Math.pow(cathet\_2, 2)

== Math.pow(hypotenuse, 2);

}

public Float getCathet\_1() {

return cathet\_1;

}

public void setCathet\_1(Float cathet\_1) {

this.cathet\_1 = cathet\_1;

}

public Float getCathet\_2() {

return cathet\_2;

}

public void setCathet\_2(Float cathet\_2) {

this.cathet\_2 = cathet\_2;

}

public Float getHypotenuse() {

return hypotenuse;

}

public void setHypotenuse(Float hypotenuse) {

this.hypotenuse = hypotenuse;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

Triangle triangle = (Triangle) o;

return cathet\_1.equals(triangle.cathet\_1) && cathet\_2.equals(triangle.cathet\_2) && hypotenuse.equals(triangle.hypotenuse);

}

@Override

public String toString() {

return "Катет №1 = " + cathet\_1 +

", Катет №2 = " + cathet\_2 +

", Гипотенуза = " + hypotenuse;

}

}

**Bus.java:**

package Model;

public class Bus {

private Long id;

private String FIO;

private String autoNumber;

private String routeNumber;

private String brand;

private int yearOperation;

private Long mileage;

private String location;

public Bus(String FIO, String autoNumber, String routeNumber, String brand, int yearOperation, Long mileage, String location) {

this.FIO = FIO;

this.autoNumber = autoNumber;

this.routeNumber = routeNumber;

this.brand = brand;

this.yearOperation = yearOperation;

this.mileage = mileage;

this.location = location;

}

public Bus() {

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getFIO() {

return FIO;

}

public void setFIO(String FIO) {

this.FIO = FIO;

}

public String getAutoNumber() {

return autoNumber;

}

public void setAutoNumber(String autoNumber) {

this.autoNumber = autoNumber;

}

public String getRouteNumber() {

return routeNumber;

}

public void setRouteNumber(String routeNumber) {

this.routeNumber = routeNumber;

}

public String getBrand() {

return brand;

}

public void setBrand(String brand) {

this.brand = brand;

}

public int getYearOperation() {

return yearOperation;

}

public void setYearOperation(int yearOperation) {

this.yearOperation = yearOperation;

}

public Long getMileage() {

return mileage;

}

public void setMileage(Long mileage) {

this.mileage = mileage;

}

public String getLocation() {

return location;

}

public void setLocation(String location) {

this.location = location;

}

@Override

public String toString() {

return "Bus{" +

"FIO='" + FIO + '\'' +

", autoNumber='" + autoNumber + '\'' +

", routeNumber='" + routeNumber + '\'' +

", brand='" + brand + '\'' +

", yearOperation=" + yearOperation +

", mileage=" + mileage +

", location='" + location + '\'' +

'}';

}

public void print() {

System.out.println(this.toString());

}

}

**LocationStatus.java:**package Model.enums;

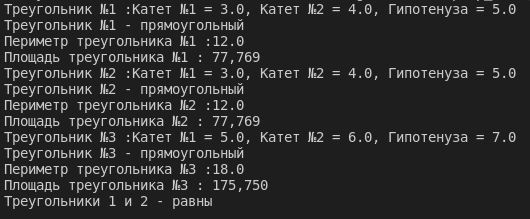
public enum LocationStatus {

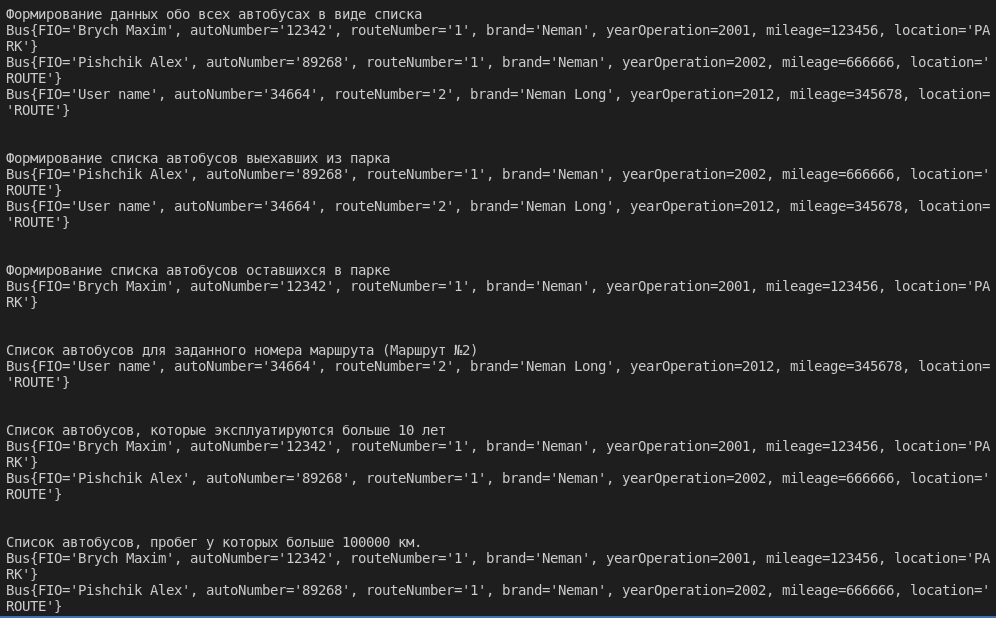
PARK,

ROUTE

}

Тестирование:





Вывод: научиться создавать и использовать классы в программах на языке программирования Java.