Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Рязанский государственный радиотехнический университет»  
(ФГБОУ ВО «РГРТУ», РГРТУ)

Кафедра «Вычислительная и прикладная математика» (ВПМ)

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 8

по дисциплине

**«ПРОГРАММИРОВАНИЕ НА JAVA»**

Выполнил:

студент группы 643

Паршина Анна Романовна

Проверил:

Пруцков Александр Викторович,

д-р техн. наук, профессор кафедры ВПМ

Рязань 2019

# Задание

Треугольники. В БД хранятся треугольники и их координаты на плоскости.

1. Вывести треугольник, площадь которого наиболее приближена к заданной.

2. Вывести треугольники, сумма площадей которых наиболее приближена к заданной.

3. Вывести треугольники, которые помещаются в окружность заданного радиуса.

# Основные классы, реализующие задание

## Класс Runner

package ru.rsreu.parshina0813;

import java.sql.SQLException;

import java.util.List;

import ru.rsreu.parshina0813.datadao.ClientDAO;

import ru.rsreu.parshina0813.datadao.DAOFactory;

import ru.rsreu.parshina0813.datadao.DBType;

import ru.rsreu.parshina0813.dboracle.OracleClientDAO;

import ru.rsreu.parshina0813.triangle.Triangle;

import ru.rsreu.parshina0813.triangle.Triangles;

public class Runner {

private Runner() {

}

public static void main(String[] args) {

StringBuilder result = new StringBuilder();

DAOFactory factory = DAOFactory.getInstance(DBType.ORACLE);

ClientDAO clientDAO = factory.getClientDAO();

try {

List<String> client = clientDAO.selectInformationClient();

result.append(Resourcer.getString("message.client"));

result.append(client).append("\n");

} catch (SQLException e) {

e.printStackTrace();

}

try {

result.append(Resourcer.getString("message.triangles"));

Triangles triangles = new Triangles();

triangles.setTriangles(OracleClientDAO.getTriangles());

result.append(triangles);

} catch (SQLException e) {

e.printStackTrace();

}

try {

result.append(Resourcer.getString("message.findTriangleNearestSquare"));

result.append(OracleClientDAO.findTriangleNearestSquare()).append("\n");

} catch (SQLException e) {

e.printStackTrace();

}

try {

result.append(Resourcer.getString("message.findTriangleInscribedCiracle"));

result.append(OracleClientDAO.findTriangleInscribedCiracle()).append("\n");

} catch (SQLException e) {

e.printStackTrace();

}

System.out.println(result);

}

}

}

## Класс OracleClientDAO

package ru.rsreu.parshina0813.dboracle;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import ru.rsreu.parshina0813.datadao.ClientDAO;

import ru.rsreu.parshina0813.Resourcer;

public class OracleClientDAO implements ClientDAO {

private static Connection connection;

public OracleClientDAO(Connection connection) {

OracleClientDAO.connection = connection;

}

public static String getTriangles() throws SQLException {

StringBuilder resultList = new StringBuilder();

PreparedStatement pstmt = null;

ResultSet result = null;

pstmt = connection.prepareStatement(Resourcer.getString("sql.quaery.getTriangles"));

result = pstmt.executeQuery();

while (result.next()) {

resultList.append(result.getString(1)).append(" ");

resultList.append(result.getString(2)).append(" ");

resultList.append(result.getString(3)).append(" ");

resultList.append(result.getString(4)).append("\n");

}

return resultList.toString();

}

public static String findTriangleNearestSquare() throws SQLException {

StringBuilder resultList = new StringBuilder();

PreparedStatement pstmt = null;

ResultSet result = null;

pstmt = connection.prepareStatement(Resourcer.getString("sql.quaery.findTriangleNearestSquare"));

result = pstmt.executeQuery();

while (result.next()) {

resultList.append(result.getString(1));

}

return resultList.toString();

}

public static String findTriangleInscribedCiracle() throws SQLException {

StringBuilder resultList = new StringBuilder();

PreparedStatement pstmt = null;

ResultSet result = null;

pstmt = connection.prepareStatement(Resourcer.getString("sql.quaery.findTriangleInscribedCiracle"));

result = pstmt.executeQuery();

while (result.next()) {

resultList.append(result.getString(1)).append(" ");

}

return resultList.toString();

}

@Override

public List<String> selectInformationClient() throws SQLException {

List<String> resultList = new ArrayList<String>();

PreparedStatement pstmt = null;

ResultSet result = null;

pstmt = connection.prepareStatement(Resourcer.getString("sql.query.clientDB"));

result = pstmt.executeQuery();

while (result.next()) {

resultList.add(result.getString(1));

resultList.add(result.getString(2));

}

return resultList;

}

}

## Класс OracleDBDaoFactory

package ru.rsreu.parshina0813.dboracle;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.Locale;

import ru.rsreu.parshina0813.datadao.ClientDAO;

import ru.rsreu.parshina0813.datadao.DAOFactory;

import ru.rsreu.parshina0813.Resourcer;

public class OracleDBDaoFactory extends DAOFactory {

private static volatile OracleDBDaoFactory instance;

private Connection connection;

private OracleDBDaoFactory() {

}

public static OracleDBDaoFactory getInstance() throws ClassNotFoundException, SQLException {

OracleDBDaoFactory factory = instance;

if (instance == null) {

synchronized (OracleDBDaoFactory.class) {

factory = new OracleDBDaoFactory();

instance = factory;

factory.connected();

}

}

return factory;

}

private void connected() throws ClassNotFoundException, SQLException {

Class.forName("oracle.jdbc.driver.OracleDriver");

Locale.setDefault(Locale.PRC);

StringBuilder result = new StringBuilder();

String url = "jdbc:oracle:thin:@localhost:1521:XE";

String user = "ANNA";

String password = "111";

connection = DriverManager.getConnection(url, user, password);

result.append(Resourcer.getString("message.connect"));

System.out.println(result);

}

@Override

public ClientDAO getClientDAO() {

return new OracleClientDAO(connection);

}

}

## Класс Point

package ru.rsreu.parshina0813.triangle;

import ru.rsreu.parshina0813.Resourcer;

public class Point {

private int x;

private int y;

public Point(int x, int y) {

this.x = x;

this.y = y;

}

public int getX() {

return x;

}

public void setX(int x) {

this.x = x;

}

public int getY() {

return y;

}

public void setY(int y) {

this.y = y;

}

@Override

public String toString() {

return String.format(Resourcer.getString("format.point"), this.x, this.y);

}

}

## Класс ComparatorTwoFilds

package ru.rsreu.parshina0813.triangle;

import java.util.ArrayList;

import java.util.List;

public class Triangles {

private List<Triangle> triangles;

public Triangles() {

triangles = null;

}

public void setTriangles(String inputTriangle) {

triangles = new ArrayList<Triangle>();

String[] triangle = inputTriangle.split("\n");

for (int i = 0; i < triangle.length; i += 3) {

String[] point1 = triangle[i].split(" ");

String[] point2 = triangle[i + 1].split(" ");

String[] point3 = triangle[i + 2].split(" ");

triangles.add(new Triangle(Integer.parseInt(point1[0]),

new Point(Integer.parseInt(point1[1]), Integer.parseInt(point1[2])),

new Point(Integer.parseInt(point2[1]), Integer.parseInt(point2[2])),

new Point(Integer.parseInt(point3[1]), Integer.parseInt(point3[2]))));

}

}

@Override

public String toString() {

StringBuilder result = new StringBuilder();

for (Triangle item : triangles) {

result.append(item).append("\n");

}

return result.toString();

}

}