**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 5**

по дисциплине:

«Визуальные средства разработки программных приложений»

на тему:

«Разработка приложений с графическим интерфейсом *Java*»

Выполнил: студент гр. ИТП-31

Коркуц С. И.

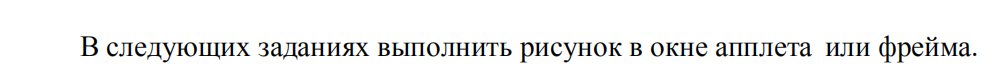
Принял: ассистент

Гуменников Е.Д.

Гомель 2020

**Цель работы:** изучить создание приложений с графическим интерфейсом в *Java*.

**Задание:** Вариант 7. Условия задания на рисунке 1.



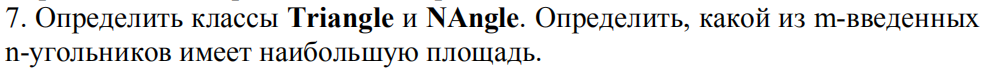


Рисунок 1 – Вариант задания

Структура созданных классов на рисунке 2.

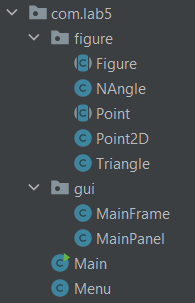


Рисунок 2 – Структура созданных классов

Был создан абстрактный класс точка, после чего от него был унаследован класс двумерная точка.

Далее был создан абстрактный класс фигура, которая хранит в себе имя фигуры и метод отрисовки фигуры. От него был унаследован класс многоугольник. Треугольник был унаследован от многоугольника, с ограничением для количества вершин равное трём.

Для графического интерфейса было написано два класса. Один для панели рисования, один для основного графического окна.

Далее был написан класс для работы с меню.

Результат выполнения программы на рисунках 3 – .

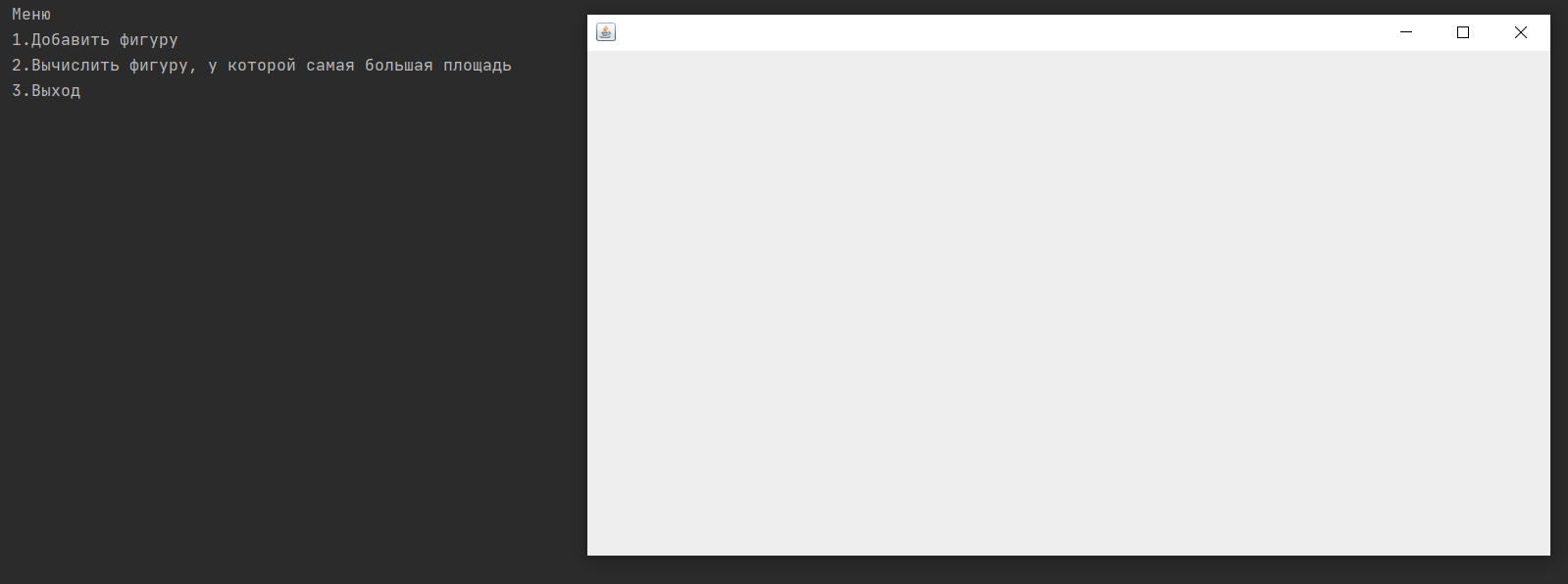


Рисунок 3 – Начало выполнения программы

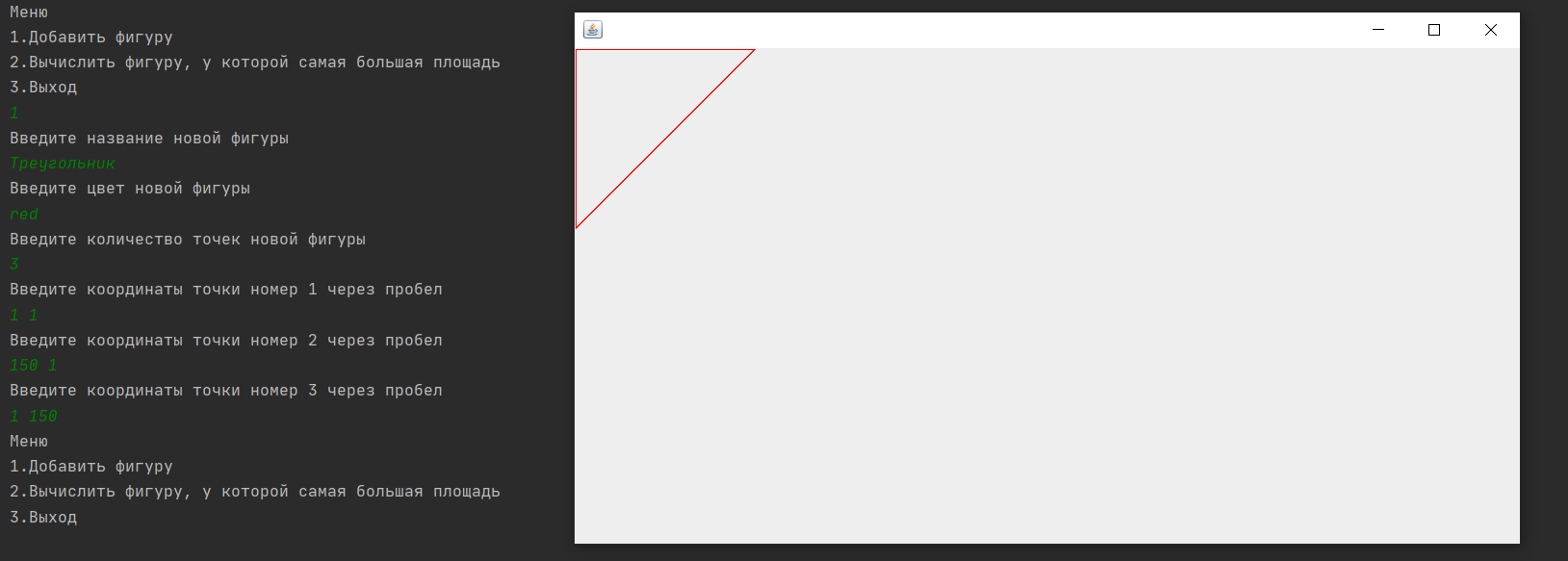


Рисунок 4 – Создание новой фигуры

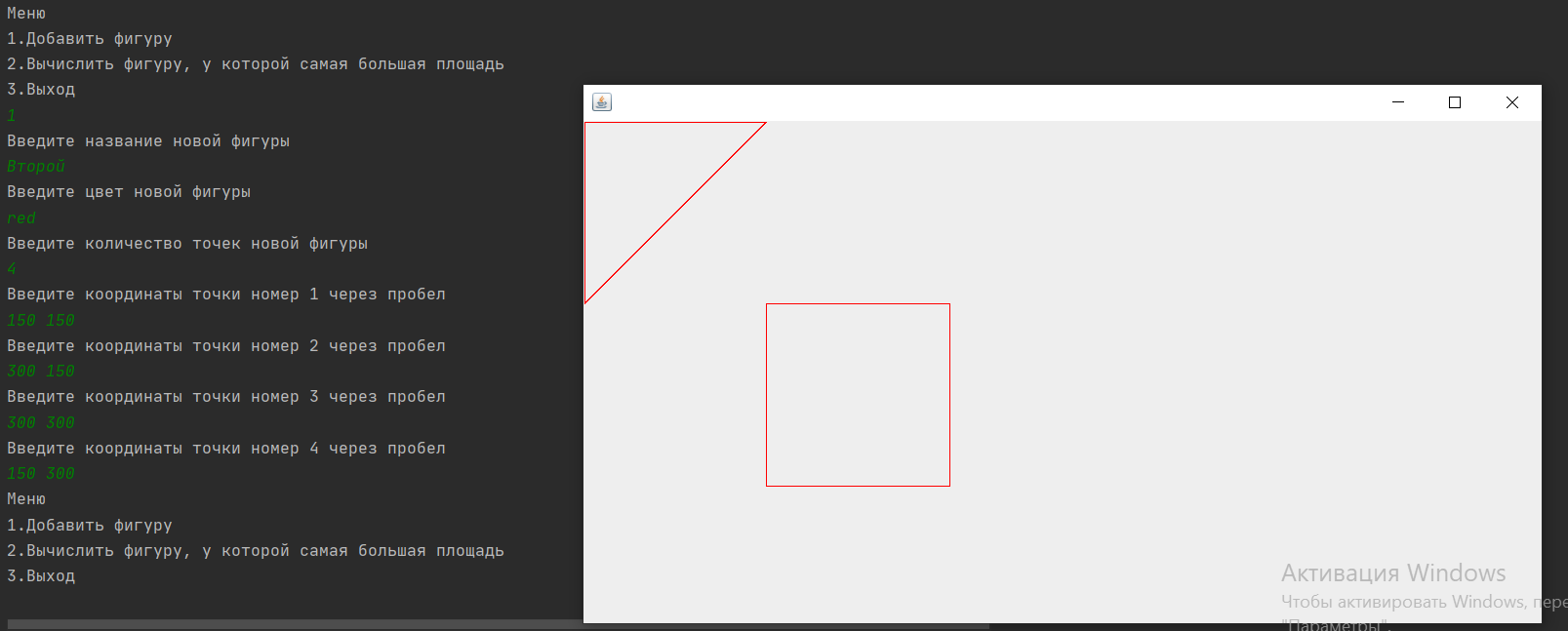


Рисунок 5 – Создание второй фигуры

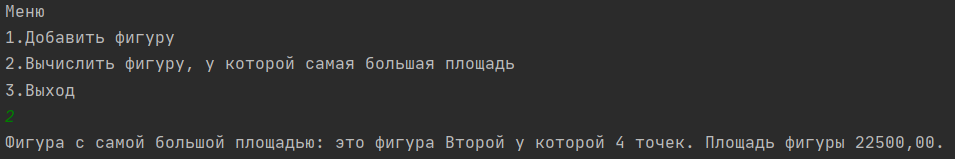


Рисунок 6 – Вычисление самой большой площади

**Вывод:** Было создано графическое приложение для работы с многоугольниками.

**ПРИЛОЖЕНИЕ А**

Листинг созданных классов

package com.lab5.figure;  
  
import java.util.List;  
  
public abstract class Point {  
 private double[] coordsVector;  
  
 protected Point(double[] coordsVector) {  
 this.coordsVector = coordsVector;  
 }  
  
 public double getDistanceFrom(Point point) throws Exception {  
 if(point.coordsVector.length != this.coordsVector.length)  
 throw new Exception("Эти точки находятся в разных пространствах");  
 double accum = 0;  
 for(int i = 0; i < this.coordsVector.length; i++) {  
 accum += Math.pow((this.coordsVector[i] - point.coordsVector[i]), 2);  
 }  
 return Math.sqrt(accum);  
 }  
  
 public double[] getCoordinatesVector() {  
 return coordsVector;  
 }  
}

package com.lab5.figure;  
  
import java.awt.\*;  
  
public abstract class Figure {  
 String name;  
 Color color;  
  
 public Figure(String name, Color color) {  
 this.name = name;  
 this.color = color;  
 }  
  
 public abstract void Draw(Graphics g);  
 public abstract double getArea();  
}

package com.lab5.figure;  
  
public class Point2D extends Point {  
 private double x, y;  
  
 public Point2D(double x, double y) {  
 super(new double[] {x, y});  
 this.x = x;  
 this.y = y;  
 }  
  
 public double getX() {  
 return x;  
 }  
  
 public double getY() {  
 return y;  
 }  
}

package com.lab5.figure;  
  
import java.awt.\*;  
import java.util.List;  
  
public class NAngle extends Figure {  
 private List<Point2D> points;  
  
 public NAngle(List<Point2D> points, String name, Color color) throws Exception {  
 super(name, color);  
 if(points.size() < 3)  
 throw new Exception("У многоугольника минимум 3 точки");  
 this.points = points;  
 }  
  
 @Override  
 public void Draw(Graphics g) {  
 var g2d = (Graphics2D)g;  
 g2d.setColor(color);  
 g2d.drawLine((int)points.get(0).getX(), (int)points.get(0).getY(),  
 (int)points.get(points.size() - 1).getX(), (int)points.get(points.size() - 1).getY());  
 for(int i = 1; i < points.size(); i++) {  
 g2d.drawLine((int)points.get(i).getX(), (int)points.get(i).getY(),  
 (int)points.get(i - 1).getX(), (int)points.get(i - 1).getY());  
 }  
 }  
  
 @Override  
 public double getArea() {  
 //Площадь n угольника вычисляется по формуле Гаусса  
 double accum = 0;  
 for(int i = 0; i < points.size() - 1; i++) {  
 accum += points.get(i).getX() \* points.get(i + 1).getY();  
 }  
 accum += points.get(points.size() - 1).getX() \* points.get(0).getY();  
 for(int i = 0; i < points.size() - 1; i++) {  
 accum -= points.get(i + 1).getX() \* points.get(i).getY();  
 }  
 accum -= points.get(0).getX() \* points.get(points.size() - 1).getY();  
 return 0.5 \* Math.abs(accum);  
 }  
  
 @Override  
 public String toString() {  
 return String.format("это фигура %s у которой %d точек. Площадь фигуры %.2f.", name, points.size(), getArea());  
 }  
}

package com.lab5.figure;  
  
import java.awt.\*;  
import java.util.List;  
  
public class Triangle extends NAngle {  
 public Triangle(List<Point2D> points, String name, Color color) throws Exception {  
 super(points, name, color);  
 if(points.size() != 3)  
 throw new Exception("У треугольника три точки");  
 }  
}

package com.lab5.gui;  
  
import com.lab5.figure.Figure;  
  
import javax.swing.\*;  
import java.util.List;  
  
public class MainFrame extends JFrame {  
  
 MainPanel panel;  
  
 public MainFrame(List<Figure> figures) {  
 this.setVisible(true);  
 this.setSize(800, 450);  
 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
  
 panel = new MainPanel(figures);  
 add(panel);  
 }  
  
 public void updateDrawPanel() {  
 panel.revalidate();  
 panel.repaint();  
 }  
}

package com.lab5.gui;  
  
import com.lab5.figure.Figure;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.util.List;  
  
public class MainPanel extends JPanel {  
  
 List<Figure> figures;  
  
 public MainPanel(List<Figure> figures) {  
 this.figures = figures;  
 this.setVisible(true);  
 }  
  
 @Override  
 protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 for (Figure figure :  
 figures) {  
 figure.Draw(g);  
 }  
 }  
}

package com.lab5;  
  
import com.lab5.figure.Figure;  
import com.lab5.figure.NAngle;  
import com.lab5.figure.Point2D;  
import com.lab5.gui.MainFrame;  
  
import java.awt.\*;  
import java.lang.reflect.Field;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class Menu {  
 Scanner in;  
 List<Figure> figures;  
 MainFrame frame;  
  
 public Menu() {  
 in = new Scanner(System.in);  
 figures = new ArrayList<>();  
 frame = new MainFrame(figures);  
 }  
  
 public void menu() {  
 int result;  
 do {  
 System.out.println("Меню");  
 printMenuItems();  
 result = in.nextInt();  
 switch (result) {  
 case 1 -> addFigureAndUpdateDrawPanel();  
 case 2 -> printMaxAreaFigure();  
 }  
 }  
 while (result != 3);  
 }  
  
 public void printMenuItems() {  
 System.out.println("1.Добавить фигуру");  
 System.out.println("2.Вычислить фигуру, у которой самая большая площадь");  
 System.out.println("3.Выход");  
 }  
  
 public void addFigureAndUpdateDrawPanel() {  
 String name;  
 System.out.println("Введите название новой фигуры");  
 name = in.next();  
 String strColor;  
 System.out.println("Введите цвет новой фигуры");  
 strColor = in.next();  
 Color color;  
 try {  
 color = getColorFromString(strColor);  
 }  
 catch (Exception exception) {  
 System.out.println(exception.getMessage());  
 return;  
 }  
 int pointsQuantity;  
 System.out.println("Введите количество точек новой фигуры");  
 pointsQuantity = in.nextInt();  
 if(pointsQuantity < 3) {  
 System.out.println("Не может быть меньше 3 точек");  
 return;  
 }  
 List<Point2D> points = new ArrayList<>();  
 for(int i = 0; i < pointsQuantity; i++) {  
 System.out.println(String.format("Введите координаты точки номер %d через пробел", i + 1));  
 double x, y;  
 x = in.nextDouble();  
 y = in.nextDouble();  
 points.add(new Point2D(x, y));  
 }  
 try {  
 figures.add(new NAngle(points, name, color));  
 }  
 catch (Exception exception) {  
 System.out.println("Неправильная фигура");  
 System.out.println(exception.getMessage());  
 }  
 frame.updateDrawPanel();  
 }  
  
 private Color getColorFromString(String colorStr) throws Exception {  
 Color color;  
 Field field = Color.class.getField(colorStr);  
 return (Color)field.get(null);  
 }  
  
 public void printMaxAreaFigure() {  
 Figure maxAreaFigure = null;  
 double maxArea = 0;  
 for (Figure figure :  
 figures) {  
 if(figure.getArea() > maxArea) {  
 maxArea = figure.getArea();  
 maxAreaFigure = figure;  
 }  
 }  
 System.out.println("Фигура с самой большой площадью: " + maxAreaFigure);  
 }  
  
}

package com.lab5;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Menu menu = new Menu();  
 menu.menu();  
 }  
}