**1.**

//Пищик Е.С. М8О-206Б-19 Лабораторная №3 Вариант 16

//8-уголник, треугольник, квадрат

#include <iostream>

#include <string>

#include <cmath>

#include <string>

#include <fstream>

#include <vector>

#include "structs.hpp"

class Figure

{

    public:

    virtual Point center() = 0;

    virtual void print\_figure() const = 0;

    virtual double area() = 0;

    virtual ~Figure() {}

};

//Наследуемый класс от Figure

//Треугольник задается 3-мя точками

class Triangle : public Figure

{

    Point p1\_;

    Point p2\_;

    Point p3\_;

    public:

    //Геттеры

    Point& get\_point1() {return p1\_;}

    Point& get\_point2() {return p2\_;}

    Point& get\_point3() {return p3\_;}

    //Конструкторы

    Triangle(){}

    Triangle(Point const& p1, Point const& p2, Point const& p3) : p1\_(p1), p2\_(p2), p3\_(p3) {}

    //Методы

    //Нахождение точки центра фигуры

    Point center(){ return Point((this->p1\_.x\_ + this->p2\_.x\_ + this->p3\_.x\_)/3, (this->p1\_.y\_ + this->p2\_.y\_ + this->p3\_.y\_)/3); }

    //Вычисление площади фигуры

    double area()

    {

        double s = (this->p1\_.x\_ - this->p3\_.x\_)\*(this->p2\_.y\_ - this->p3\_.y\_) - (this->p1\_.y\_ - this->p3\_.y\_)\*(this->p2\_.x\_ - this->p3\_.x\_)/2;

        return s < 0 ? -s : s;

    }

    //Печать координат вершин на поток вывода

    void print\_figure() const

    {

        std::cout << std::endl;

        std::cout << "Triangle:" << std::endl;

        std::cout << "First point: (" << this->p1\_.x\_ << "," << this->p1\_.y\_ << ")" << std::endl;

        std::cout << "Second point: (" << this->p2\_.x\_ << "," << this->p2\_.y\_ << ")" << std::endl;

        std::cout << "Third point: (" << this->p3\_.x\_ << "," << this->p3\_.y\_ << ")" << std::endl;

    }

};

//Наследуемый класс от Figure

//Квадрат задается 2 точками одной из диагоналей

class Square : public Figure

{

    Point p1\_;

    Point p2\_;

    public:

    //Геттеры

    Point& get\_point1() {return p1\_;}

    Point& get\_point2() {return p2\_;}

    //Конструктор

    Square(){}

    Square(Point const& p1, Point const& p2) : p1\_(p1), p2\_(p2) {}

    //Методы

    //Нахождение точки центра фигуры

    Point center(){ return Point((p1\_.x\_ + p2\_.x\_)/2, (p1\_.y\_ + p2\_.y\_)/2); }

    //Вычисление площади фигуры

    double area() { return pow(p1\_.x\_ - p2\_.x\_, 2); }

    //Печать координат вершин на поток вывода

    void print\_figure() const

    {

        std::cout << std::endl;

        std::cout << "Square:" << std::endl;

        std::cout << "First point: (" << this->p1\_.x\_ << "," << this->p1\_.y\_ << ")" << std::endl;

        std::cout << "Second point: (" << this->p2\_.x\_ << "," << this->p2\_.y\_ << ")" << std::endl;

    }

};

//Наследуемый класс от Figure

//Восьмиугольник задается 8 точками

class Octagon : public Figure

{

    Point p1\_;

    Point p2\_;

    Point p3\_;

    Point p4\_;

    Point p5\_;

    Point p6\_;

    Point p7\_;

    Point p8\_;

    public:

    //Геттеры

    Point& get\_point1() {return p1\_;}

    Point& get\_point2() {return p2\_;}

    Point& get\_point3() {return p3\_;}

    Point& get\_point4() {return p4\_;}

    Point& get\_point5() {return p5\_;}

    Point& get\_point6() {return p6\_;}

    Point& get\_point7() {return p7\_;}

    Point& get\_point8() {return p8\_;}

    //Конструктор

    Octagon(){}

    Octagon(Point const& p1, Point const& p2, Point const& p3, Point const& p4, Point const& p5, Point const& p6, Point const& p7, Point const& p8) :

    p1\_(p1), p2\_(p2), p3\_(p3), p4\_(p4), p5\_(p5), p6\_(p6), p7\_(p7), p8\_(p8) {}

    //Методы

    //Нахождение точки центра фигуры

    Point center()

    {

        double x\_center, y\_center = 0;

        x\_center = (this->p1\_.x\_ + this->p2\_.x\_ + this->p3\_.x\_ + this->p4\_.x\_ + this->p5\_.x\_ + this->p6\_.x\_ + this->p7\_.x\_ + this->p8\_.x\_)/8;

        y\_center = (this->p1\_.y\_ + this->p2\_.y\_ + this->p3\_.y\_ + this->p4\_.y\_ + this->p5\_.y\_ + this->p6\_.y\_ + this->p7\_.y\_ + this->p8\_.y\_)/8;

        return Point(x\_center, y\_center);

    }

    //Вычисление площади фигуры

    double area()

    {

        Point c = this->center();

        double s = 0;

        s += (p1\_.x\_\*p2\_.y\_ + p2\_.x\_\*p3\_.y\_ + p3\_.x\_\*p4\_.y\_ + p4\_.x\_\*p5\_.y\_ + p5\_.x\_\*p6\_.y\_ + p6\_.x\_\*p7\_.y\_ + p7\_.x\_\*p8\_.y\_ + p8\_.x\_\*p1\_.y\_);

        s -= (p2\_.x\_\*p1\_.y\_ + p3\_.x\_\*p2\_.y\_ + p4\_.x\_\*p3\_.y\_ + p5\_.x\_\*p4\_.y\_ + p6\_.x\_\*p5\_.y\_ + p7\_.x\_\*p6\_.y\_ + p8\_.x\_\*p7\_.y\_ + p1\_.x\_\*p8\_.y\_);

        return s > 0 ? s/2 : -(s/2);

    }

    //Печать координат вершин на поток вывода

    void print\_figure() const

    {

        std::cout << std::endl;

        std::cout << "Octagon:" << std::endl;

        std::cout << "First point: (" << this->p1\_.x\_ << "," << this->p1\_.y\_ << ")" << std::endl;

        std::cout << "Second point: (" << this->p2\_.x\_ << "," << this->p2\_.y\_ << ")" << std::endl;

        std::cout << "Third point: (" << this->p3\_.x\_ << "," << this->p3\_.y\_ << ")" << std::endl;

        std::cout << "Fourth point: (" << this->p4\_.x\_ << "," << this->p4\_.y\_ << ")" << std::endl;

        std::cout << "Fifth point: (" << this->p5\_.x\_ << "," << this->p5\_.y\_ << ")" << std::endl;

        std::cout << "Sixth point: (" << this->p6\_.x\_ << "," << this->p6\_.y\_ << ")" << std::endl;

        std::cout << "Seventh point: (" << this->p7\_.x\_ << "," << this->p7\_.y\_ << ")" << std::endl;

        std::cout << "Eighth point: (" << this->p8\_.x\_ << "," << this->p8\_.y\_ << ")" << std::endl;

    }

};

//Для ввода через cin собственных классов

//Вводится 16 чисел, координаты 8 точек для восьмиугольника

std::istream& operator>>(std::istream &is, Octagon& ogon)

{

    std::string p11, p12, p21, p22, p31, p32, p41, p42, p51, p52, p61, p62, p71, p72, p81, p82;

    while(true)

    {

        try

        {

            is >> p11 >> p12 >> p21 >> p22 >> p31 >> p32 >> p41 >> p42 >> p51 >> p52 >> p61 >> p62 >> p71 >> p72 >> p81 >> p82;

            ogon.get\_point1().x\_ = std::stod(p11);

            ogon.get\_point1().y\_ = std::stod(p12);

            ogon.get\_point2().x\_ = std::stod(p21);

            ogon.get\_point2().y\_ = std::stod(p22);

            ogon.get\_point3().x\_ = std::stod(p31);

            ogon.get\_point3().y\_ = std::stod(p32);

            ogon.get\_point4().x\_ = std::stod(p41);

            ogon.get\_point4().y\_ = std::stod(p42);

            ogon.get\_point5().x\_ = std::stod(p51);

            ogon.get\_point5().y\_ = std::stod(p52);

            ogon.get\_point6().x\_ = std::stod(p61);

            ogon.get\_point6().y\_ = std::stod(p62);

            ogon.get\_point7().x\_ = std::stod(p71);

            ogon.get\_point7().y\_ = std::stod(p72);

            ogon.get\_point8().x\_ = std::stod(p81);

            ogon.get\_point8().y\_ = std::stod(p82);

            return is;

        }

        catch(const std::exception& ex)

        {

            std::cerr << "Incorrect data, try again" << std::endl;

            std::cerr << ex.what() << std::endl;

        }

    }

}

//Вводится 4 числа координаты двух диагональных точек для квадрата

std::istream& operator>>(std::istream &is, Square& sqr)

{

    std::string p11, p12, p21, p22;

    while(true)

    {

        try

        {

            is >> p11 >> p12 >> p21 >> p22;

            sqr.get\_point1().x\_ = std::stod(p11);

            sqr.get\_point1().y\_ = std::stod(p12);

            sqr.get\_point2().x\_ = std::stod(p21);

            sqr.get\_point2().y\_ = std::stod(p22);

            return is;

        }

        catch(const std::exception& ex)

        {

            std::cerr << "Incorrect data, try again" << std::endl;

            std::cerr << ex.what() << std::endl;

        }

    }

}

//Вводится 6 чисел по 2 числа на каждую вершину трейгольника

std::istream& operator>>(std::istream &is, Triangle& tr)

{

    std::string p11, p12, p21, p22, p31, p32;

    while(true)

    {

        try

        {

            is >> p11 >> p12 >> p21 >> p22 >> p31 >> p32;

            tr.get\_point1().x\_ = std::stod(p11);

            tr.get\_point1().y\_ = std::stod(p12);

            tr.get\_point2().x\_ = std::stod(p21);

            tr.get\_point2().y\_ = std::stod(p22);

            tr.get\_point3().x\_ = std::stod(p31);

            tr.get\_point3().y\_ = std::stod(p32);

            return is;

        }

        catch(const std::exception& ex)

        {

            std::cerr << "Incorrect data, try again" << std::endl;

            std::cerr << ex.what() << std::endl;

        }

    }

}

//Функция вычисления площади в векторе

double area\_in\_vector(std::vector<Figure\*> figures)

{

    double sum = 0;

    std::vector<Figure\*>::iterator iter = figures.begin();

    while(iter != figures.end())

    {

        //\*iter элемент на который указывает iter, т.е Figure\*

        sum += (\*iter)->area();

        ++iter;

    }

    return sum;

}

//Удаление эл-та по индексу

void delete\_el(std::vector<Figure\*> & figures, int i = 0)

{

    figures.erase(figures.begin() + i);

}

//Тестирующая функция для треугольника

void TestTriangle(double p11, double p12, double p21, double p22, double p31, double p32)

{

    Point p1(p11, p12);

    Point p2(p21, p22);

    Point p3(p31, p32);

    Triangle tr(p1, p2, p3);

    tr.print\_figure();

    std::cout << "Area = " << tr.area() << std::endl;

    std::cout << "Center x = " << tr.center().x\_ << ", y = " << tr.center().y\_ << std::endl;

}

//Тестирующая функция для квадрата

void TestSquare(double p11, double p12, double p21, double p22)

{

    Point p1(p11, p12);

    Point p2(p21, p22);

    Square sqr(p1, p2);

    sqr.print\_figure();

    std::cout << "Area = " << sqr.area() << std::endl;

    std::cout << "Center x = " << sqr.center().x\_ << ", y = " << sqr.center().y\_ << std::endl;

}

//Тестирующая функция для 8-миугольника

void TestOctagon(double p11, double p12, double p21, double p22, double p31, double p32, double p41, double p42,

double p51, double p52, double p61, double p62, double p71, double p72, double p81, double p82)

{

    //Координаты передаются по круговому обходу

    Point p1(p11, p12);

    Point p2(p21, p22);

    Point p3(p31, p32);

    Point p4(p41, p42);

    Point p5(p51, p52);

    Point p6(p61, p62);

    Point p7(p71, p72);

    Point p8(p81, p82);

    Octagon ogon(p1, p2, p3, p4, p5, p6, p7, p8);

    ogon.print\_figure();

    std::cout << "Area = " << ogon.area() << std::endl;

    std::cout << "Center x = " << ogon.center().x\_ << ", y = " << ogon.center().y\_ << std::endl;

}

//Тестирующая функция для cin с моими классами

void TestCin()

{

    std::cout << "Enter 16 numbers, coordinates 8 points, in the correct order (in a circle) for octagon:" << std::endl;

    Octagon ogon;

    std::cin >> ogon;

    ogon.print\_figure();

    std::cout << "Area = " << ogon.area() << std::endl;

    std::cout << "Center x = " << ogon.center().x\_ << ", y = " << ogon.center().y\_ << std::endl;

    std::cout << "Enter 4 numbers - 2 pairs diagonal coordinates for square:"<< std::endl;

    Square sqr;

    std::cin >> sqr;

    sqr.print\_figure();

    std::cout << "Area = " << sqr.area() << std::endl;

    std::cout << "Center x = " << sqr.center().x\_ << ", y = " << sqr.center().y\_ << std::endl;

    std::cout << "Enter 6 numbers - 3 pairs coordinates for triangle:" << std::endl;

    Triangle tr;

    std::cin >> tr;

    tr.print\_figure();

    std::cout << "Area = " << tr.area() << std::endl;

    std::cout << "Center x = " << tr.center().x\_ << ", y = " << tr.center().y\_ << std::endl;

}

//Тестирующая функция для вектора

void TestVector()

{

    //Вектор хранящий элементы типа Figure\*

    std::vector<Figure\*> figures;

    std::cout << "Enter 16 numbers, coordinates 8 points, in the correct order (in a circle) for octagon:" << std::endl;

    Octagon ogon;

    std::cin >> ogon;

    std::cout << "Enter 4 numbers - 2 pairs diagonal coordinates for square:"<< std::endl;

    Square sqr;

    std::cin >> sqr;

    std::cout << "Enter 6 numbers - 3 pairs coordinates for triangle:" << std::endl;

    Triangle tr;

    std::cin >> tr;

    figures.push\_back(&ogon);

    figures.push\_back(&sqr);

    figures.push\_back(&tr);

    //Итератор на начало вектора

    std::vector<Figure\*>::iterator iter = figures.begin();

    while(iter != figures.end())

    {

        //\*iter элемент на который указывает iter, т.е Figure\*

        (\*iter)->print\_figure();

        std::cout << "Area = " << (\*iter)->area() << std::endl;

        std::cout << "Center x = " << (\*iter)->center().x\_ << ", y = " << (\*iter)->center().y\_ << std::endl;

        ++iter;

    }

    std::cout << "Summary area in vector: " << area\_in\_vector(figures) << std::endl;

    //удаляем элемент с индексом 0

    delete\_el(figures, 0);

    iter = figures.begin();

    while(iter != figures.end())

    {

        //\*iter элемент на который указывает iter, т.е Figure\*

        (\*iter)->print\_figure();

        std::cout << "Area = " << (\*iter)->area() << std::endl;

        std::cout << "Center x = " << (\*iter)->center().x\_ << ", y = " << (\*iter)->center().y\_ << std::endl;

        ++iter;

    }

    std::cout << "Summary area in vector after delete element by index: " << area\_in\_vector(figures) << std::endl;

}

int main()

{

    TestCin();

    TestVector();

    std::string name = "start";

    std::cout << "Enter q or Q or quit or Quit for close programm" << std::endl;

    //Бесконечный дикл до ввода ключевого слова для выхода

    while(true)

    {

        std::cout << "Enter test file name: ";

        std::cin >> name;

        if (name == "Q" || name == "q" || name == "Quit" || name == "quit") break;

        //поток ввода из файла, имя файла хранится в string переменной name

        std::ifstream finp(name.c\_str());

        std::string s11, s12, s21, s22;

        std::string t11, t12, t21, t22, t31, t32;

        std::string p11, p12, p21, p22, p31, p32, p41, p42, p51, p52;

        std::string p61, p62, p71, p72, p81, p82;

        //Взятие 16 чисел для восьмиугольника из файла

        finp >> p11 >> p12 >> p21 >> p22 >> p31 >> p32 >> p41 >> p42 >> p51 >> p52 >> p61 >> p62 >> p71 >> p72 >> p81 >> p82;

        //Взятие 4 чисел для квадрата из файла

        finp >> s11 >> s12 >> s21 >> s22;

        //Взятие 6 чисел для треугольника из файла

        finp >> t11 >> t12 >> t21 >> t22 >> t31 >> t32;

        try

        {

            TestOctagon(std::stod(p11), std::stod(p12), std::stod(p21), std::stod(p22), std::stod(p31), std::stod(p32), std::stod(p41),

            std::stod(p42), std::stod(p51), std::stod(p52), std::stod(p61), std::stod(p62), std::stod(p71), std::stod(p72), std::stod(p81), std::stod(p82));

            TestSquare(std::stod(s11), std::stod(s12), std::stod(s21), std::stod(s22));

            TestTriangle(std::stod(t11), std::stod(t12), std::stod(t21), std::stod(t22), std::stod(t31), std::stod(t32));

        }

        catch(std::exception& ex)

        {

            std::cerr << "Incorrect data" << std::endl;

            std::cerr << ex.what() << std::endl;

        }

        finp.close();

    }

    return 0;

}

**2.** <https://github.com/Pe4enIks/OOP_Pishchik_206>

**3.**

Тесты:

1. **Сначала интерактивно вводим**

0 0 1 2 2 4 3 2 4 0 3 -2 2 -4 1 -2

0 2 2 0

0 0 1 3 2 0

0 0 1 2 2 4 3 2 4 0 3 -2 2 -4 1 -2

0 2 2 0

0 0 1 3 2 0

**Затем ввод из файла test\_1.txt**

0 0 1 2 2 4 3 2 4 0 3 -2 2 -4 1 -2

0 2 2 0

0 1 3 2 0

1. **Сначала интерактивно вводим**

0 0 1 1 2 2 3 1 4 0 3 -1 2 -2 1 -1

-7 -3 2 6

-2 -3 0 4 4 -6

0 0 1 1 2 2 3 1 4 0 3 -1 2 -2 1 -1

-7 -3 2 6

-2 -3 0 4 4 -6

**Затем ввод из файла test\_2.txt**

0 0 1 1 2 2 3 1 4 0 3 -1 2 -2 1 -1

-7 -3 2 6

-2 -3 0 4 4 -6

1. **Сначала интерактивно вводим**

-3 -3 -1 -1 1 1 3 -1 5 -3 3 -5 1 -7 -1 -5

2.5 0 5 2.5

0.75 0.5 1.5 -2 3 1.5

-3 -3 -1 -1 1 1 3 -1 5 -3 3 -5 1 -7 -1 -5

2.5 0 5 2.5

0.75 0.5 1.5 -2 3 1.5

**Затем ввод из файла test\_3.txt**

-3 -3 -1 -1 1 1 3 -1 5 -3 3 -5 1 -7 -1 -5

2.5 0 5 2.5

0.75 0.5 1.5 -2 3 1.5

**4.**

**Результат 1 теста:**

Enter 16 numbers, coordinates 8 points, in the correct order (in a circle) for octagon:

0 0 1 2 2 4 3 2 4 0 3 -2 2 -4 1 -2

Octagon:

First point: (0,0)

Second point: (1,2)

Third point: (2,4)

Fourth point: (3,2)

Fifth point: (4,0)

Sixth point: (3,-2)

Seventh point: (2,-4)

Eighth point: (1,-2)

Area = 16

Center x = 2, y = 0

Enter 4 numbers - 2 pairs diagonal coordinates for square:

0 2 2 0

Square:

First point: (0,2)

Second point: (2,0)

Area = 4

Center x = 1, y = 1

Enter 6 numbers - 3 pairs coordinates for triangle:

0 0 1 3 2 0

Triangle:

First point: (0,0)

Second point: (1,3)

Third point: (2,0)

Area = 6

Center x = 1, y = 1

Enter 16 numbers, coordinates 8 points, in the correct order (in a circle) for octagon:

0 0 1 2 2 4 3 2 4 0 3 -2 2 -4 1 -2

Enter 4 numbers - 2 pairs diagonal coordinates for square:

0 2 2 0

Enter 6 numbers - 3 pairs coordinates for triangle:

0 0 1 3 2 0

Octagon:

First point: (0,0)

Second point: (1,2)

Third point: (2,4)

Fourth point: (3,2)

Fifth point: (4,0)

Sixth point: (3,-2)

Seventh point: (2,-4)

Eighth point: (1,-2)

Area = 16

Center x = 2, y = 0

Square:

First point: (0,2)

Second point: (2,0)

Area = 4

Center x = 1, y = 1

Triangle:

First point: (0,0)

Second point: (1,3)

Third point: (2,0)

Area = 6

Center x = 1, y = 1

Summary area in vector: 26

Square:

First point: (0,2)

Second point: (2,0)

Area = 4

Center x = 1, y = 1

Triangle:

First point: (0,0)

Second point: (1,3)

Third point: (2,0)

Area = 6

Center x = 1, y = 1

Summary area in vector after delete element by index: 10

Enter q or Q or quit or Quit for close programm

Enter test file name: test\_1.txt

Octagon:

First point: (0,0)

Second point: (1,2)

Third point: (2,4)

Fourth point: (3,2)

Fifth point: (4,0)

Sixth point: (3,-2)

Seventh point: (2,-4)

Eighth point: (1,-2)

Area = 16

Center x = 2, y = 0

Square:

First point: (0,2)

Second point: (2,0)

Area = 4

Center x = 1, y = 1

Triangle:

First point: (0,0)

Second point: (1,3)

Third point: (2,0)

Area = 6

Center x = 1, y = 1

**Результат 2 теста:**

Enter 16 numbers, coordinates 8 points, in the correct order (in a circle) for octagon:

0 0 1 1 2 2 3 1 4 0 3 -1 2 -2 1 -1

Octagon:

First point: (0,0)

Second point: (1,1)

Third point: (2,2)

Fourth point: (3,1)

Fifth point: (4,0)

Sixth point: (3,-1)

Seventh point: (2,-2)

Eighth point: (1,-1)

Area = 8

Center x = 2, y = 0

Enter 4 numbers - 2 pairs diagonal coordinates for square:

-7 -3 2 6

Square:

First point: (-7,-3)

Second point: (2,6)

Area = 81

Center x = -2.5, y = 1.5

Enter 6 numbers - 3 pairs coordinates for triangle:

-2 -3 0 4 4 -6

Triangle:

First point: (-2,-3)

Second point: (0,4)

Third point: (4,-6)

Area = 54

Center x = 0.666667, y = -1.66667

Enter 16 numbers, coordinates 8 points, in the correct order (in a circle) for octagon:

0 0 1 1 2 2 3 1 4 0 3 -1 2 -2 1 -1

Enter 4 numbers - 2 pairs diagonal coordinates for square:

-7 -3 2 6

Enter 6 numbers - 3 pairs coordinates for triangle:

-2 -3 0 4 4 -6

Octagon:

First point: (0,0)

Second point: (1,1)

Third point: (2,2)

Fourth point: (3,1)

Fifth point: (4,0)

Sixth point: (3,-1)

Seventh point: (2,-2)

Eighth point: (1,-1)

Area = 8

Center x = 2, y = 0

Square:

First point: (-7,-3)

Second point: (2,6)

Area = 81

Center x = -2.5, y = 1.5

Triangle:

First point: (-2,-3)

Second point: (0,4)

Third point: (4,-6)

Area = 54

Center x = 0.666667, y = -1.66667

Summary area in vector: 143

Square:

First point: (-7,-3)

Second point: (2,6)

Area = 81

Center x = -2.5, y = 1.5

Triangle:

First point: (-2,-3)

Second point: (0,4)

Third point: (4,-6)

Area = 54

Center x = 0.666667, y = -1.66667

Summary area in vector after delete element by index: 135

Enter q or Q or quit or Quit for close programm

Enter test file name: test\_2.txt

Octagon:

First point: (0,0)

Second point: (1,1)

Third point: (2,2)

Fourth point: (3,1)

Fifth point: (4,0)

Sixth point: (3,-1)

Seventh point: (2,-2)

Eighth point: (1,-1)

Area = 8

Center x = 2, y = 0

Square:

First point: (-7,-3)

Second point: (2,6)

Area = 81

Center x = -2.5, y = 1.5

Triangle:

First point: (-2,-3)

Second point: (0,4)

Third point: (4,-6)

Area = 54

Center x = 0.666667, y = -1.66667

**Результат 3 теста:**

Enter 16 numbers, coordinates 8 points, in the correct order (in a circle) for octagon:

-3 -3 -1 -1 1 1 3 -1 5 -3 3 -5 1 -7 -1 -5

Octagon:

First point: (-3,-3)

Second point: (-1,-1)

Third point: (1,1)

Fourth point: (3,-1)

Fifth point: (5,-3)

Sixth point: (3,-5)

Seventh point: (1,-7)

Eighth point: (-1,-5)

Area = 32

Center x = 1, y = -3

Enter 4 numbers - 2 pairs diagonal coordinates for square:

2.5 0 5 2.5

Square:

First point: (2.5,0)

Second point: (5,2.5)

Area = 6.25

Center x = 3.75, y = 1.25

Enter 6 numbers - 3 pairs coordinates for triangle:

0.75 0.5 1.5 -2 3 1.5

Triangle:

First point: (0.75,0.5)

Second point: (1.5,-2)

Third point: (3,1.5)

Area = 7.125

Center x = 1.75, y = 0

Enter 16 numbers, coordinates 8 points, in the correct order (in a circle) for octagon:

-3 -3 -1 -1 1 1 3 -1 5 -3 3 -5 1 -7 -1 -5

Enter 4 numbers - 2 pairs diagonal coordinates for square:

2.5 0 5 2.5

Enter 6 numbers - 3 pairs coordinates for triangle:

0.75 0.5 1.5 -2 3 1.5

Octagon:

First point: (-3,-3)

Second point: (-1,-1)

Third point: (1,1)

Fourth point: (3,-1)

Fifth point: (5,-3)

Sixth point: (3,-5)

Seventh point: (1,-7)

Eighth point: (-1,-5)

Area = 32

Center x = 1, y = -3

Square:

First point: (2.5,0)

Second point: (5,2.5)

Area = 6.25

Center x = 3.75, y = 1.25

Triangle:

First point: (0.75,0.5)

Second point: (1.5,-2)

Third point: (3,1.5)

Area = 7.125

Center x = 1.75, y = 0

Summary area in vector: 45.375

Square:

First point: (2.5,0)

Second point: (5,2.5)

Area = 6.25

Center x = 3.75, y = 1.25

Triangle:

First point: (0.75,0.5)

Second point: (1.5,-2)

Third point: (3,1.5)

Area = 7.125

Center x = 1.75, y = 0

Summary area in vector after delete element by index: 13.375

Enter q or Q or quit or Quit for close programm

Enter test file name: test\_3.txt

Octagon:

First point: (-3,-3)

Second point: (-1,-1)

Third point: (1,1)

Fourth point: (3,-1)

Fifth point: (5,-3)

Sixth point: (3,-5)

Seventh point: (1,-7)

Eighth point: (-1,-5)

Area = 32

Center x = 1, y = -3

Square:

First point: (2.5,0)

Second point: (5,2.5)

Area = 6.25

Center x = 3.75, y = 1.25

Triangle:

First point: (0.75,0.5)

Second point: (1.5,-2)

Third point: (3,1.5)

Area = 7.125

Center x = 1.75, y = 0

**5.** В программе создается 3 производных от абстрактного класса Figure объекта класса: Square, Triangle, Octagon, у каждого вычисляется площадь, точка центра и печатается объект на экран, также создается вектор указателей на класс Figure, и все объекты обрабатываются в цикле единым образом, также при помощи перегрузки ввода, объекты классов вводятся как cin >> any\_figure.