

Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІІІ

ЗВІТ

з виконання лабораторної роботи № 3 з

кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 1

Виконав:

студент 1-го курсу

гр. ІІІ-22 ФІОТ

Андреева Уляна Андріївна

Київ 2023

ПОСТАНОВКА ЗАДАЧІ

1. Розробити клас "трикутник на площині", який заданий координатами своїх вершин. Створити масив об'єктів даного класу. Визначити трикутник з найбільшою площею.

Код програми

main.cpp

```
#include "functions.h"
using namespace std;

int main() {
    int quantity = GetTriangleQuantity();
    Triangle *trianglesVector = InputTriangle(quantity);
    ShowTriangles(trianglesVector, quantity);
    Triangle theBiggestSquare =
FindTheBiggestSquare(trianglesVector, quantity);
    printf("\n\nThe biggest square of triangle is: %.2f\n",
theBiggestSquare.getSquare());
    printf("\t\t\tCoordinates:\n");
    printf("First point coordinates are: %.2f,%.2f\n",
theBiggestSquare.getPoint1().getX(), theBiggestSquare.getPoint1().getY());
    printf("Second point coordinates: %.2f,%.2f\n",
theBiggestSquare.getPoint2().getX(), theBiggestSquare.getPoint2().getY());
    printf("Third point coordinates: %.2f,%.2f",
theBiggestSquare.getPoint3().getX(), theBiggestSquare.getPoint3().getY());
}
```

Functions.cpp

```
#include <vector>
#include "Triangle.h"
#include <iostream>
#include <string>
#include "functions.h"
using namespace std;

int GetTriangleQuantity(){
    char ch,error;
    int quantityTriangles;

    do
    {
        error=0;
        printf("Enter quantity of triangles: "); //checking symbols
        scanf("%d%c",&quantityTriangles,&ch);
        if ((ch!='\n')||(quantityTriangles<=0))//checking buffering zone
inputting (stdin) + substracion check
        {
            error = 1;
            fflush(stdin);
            printf("Error inputting\n");
        }
        ch = 0;
    }
```

```

    }
    while(error);

    return quantityTriangles;
}

Triangle* InputTriangle(int quantity) {
    //Triangle trianglesVector[quantity];
    Triangle *trianglesVector = new Triangle[quantity];

    char ch,error;
    for (int i = 0; i < quantity; ++i) {
        cout << "Enter triangle #" << i+1 << endl;
        float x1;
        symbolChecking(error, ch, x1, "Enter 1st point x: ");
        float y1;
        symbolChecking(error, ch, y1, "Enter 1st point y: ");
        float x2;
        symbolChecking(error, ch, x2, "Enter 2nd point x: ");
        float y2;
        symbolChecking(error, ch, y2, "Enter 2nd point y: ");
        float x3;
        symbolChecking(error, ch, x3, "Enter 3d point x: ");
        float y3;
        symbolChecking(error, ch, y3, "Enter 3d point y: ");

        Point point1(x1,y1);
        Point point2(x2,y2);
        Point point3(x3,y3);
        Triangle newTriangle(point1,point2,point3);
        trianglesVector[i] = newTriangle;
    }
    return trianglesVector;
}

void symbolChecking(char error, char &ch, float &x1, std::string text) {
    do
    {
        error=0;
        cout<< text;
        scanf("%f%c",&x1,&ch);
        if (ch!='\n')//checking buffering zone inputting (stdin) +
substracion check
        {
            error = 1;
            fflush(stdin);
            printf("Error inputting\n");
        }
        ch = 0;
    }
    while(error);
}

Triangle FindTheBiggestSquare(Triangle biggestSquare[], int arraySize){
    Triangle largestTriangle = biggestSquare[0];

    for (int i = 1; i < arraySize; ++i) {
        if(biggestSquare[i].getSquare() > largestTriangle.getSquare()){
            largestTriangle = biggestSquare[i];
        }
    }
}

```

```

        return largestTriangle;
    }

void ShowTriangles(Triangle triangles[], int arraySize) {
    cout << "Array of triangles:" << endl;

    for (int i = 0; i < arraySize; ++i) {
        printf("\n-----");
        printf("\n\t\tInfo about triangle %d\n", i+1);
        printf("Square of triangle is: %.2f\n", triangles[i].getSquare());

        printf("\t\t\tCoordinates:\n");
        printf("First point coordinates are: %.2f,%.2f\n",
triangles[i].getPoint1().getX(), triangles[i].getPoint1().getY());
        printf("Second point coordinates: %.2f,%.2f\n",
triangles[i].getPoint2().getX(), triangles[i].getPoint2().getY());
        printf("Third point coordinates: %.2f,%.2f",
triangles[i].getPoint3().getX(), triangles[i].getPoint3().getY());
    }
    printf("\n-----");
}

```

Functions.h

```

#include "Triangle.h"

#ifndef LABA3_FUNCTIONS_H
#define LABA3_FUNCTIONS_H
#endif //LABA3_FUNCTIONS_H

int GetTriangleQuantity();
Triangle* InputTriangle(int quantity) ;
Triangle FindTheBiggestSquare(Triangle biggestSquare[], int arraySize);
void symbolChecking(char error, char &ch, float &x1, std::string text) ;
void ShowTriangles(Triangle triangles[], int arraySize);

```

Triangle.cpp

```

#include "Triangle.h"

Triangle::Triangle(const Point &point1, const Point &point2, const Point
&point3) : point1(point1), point2(point2),
point3(point3) {}

Triangle::Triangle(): point1(0,0),point2(0,0), point3(0,0){
}

const Point &Triangle::getPoint1() const {
    return point1;
}

const Point &Triangle::getPoint2() const {
    return point2;
}

const Point &Triangle::getPoint3() const {

```

```
        return point3;
    }
```

Triangle.h

```
#ifndef LABA3_TRIANGLE_H
#define LABA3_TRIANGLE_H

#include "Point.h"

class Triangle {
private:
    Point point1;
    Point point2;
    Point point3;

public:
    Triangle(const Point &point1, const Point &point2, const Point &point3);
    Triangle();

    const Point &getPoint1() const;

    const Point &getPoint2() const;

    const Point &getPoint3() const;

    float getSquare() const {
        return 0.5*abs((point1.getX()*point2.getY()+point2.getX()*
point3.getY() + point3.getX()*point1.getY()) -
(point1.getY()*point2.getX()+point2.getY()*point3.getX()+point3.getY()*point
1.getX()));
    }
};

#endif //LABA3_TRIANGLE_H
```

Point.cpp

```
#include "Point.h"

#include <utility>

Point::Point(float x, float y) : x(x), y(y) {}

float Point::getX() const {
    return x;
}

void Point::setX(float x) {
    Point::x = x;
}

float Point::getY() const {
    return y;
}

void Point::setY(float y) {
    Point::y = y;
}
```

Point.h

```
#ifndef LABA3_POINT_H
#define LABA3_POINT_H
#include <string>

class Point {
private:
public:
    Point(float x, float y);

private:
    float x;
    float y;
public:
    float getX() const;

    void setX(float x);

    float getY() const;

    void setY(float y);
};

#endif //LABA3_POINT_H
```

РЕЗУЛЬТАТИ ТЕСТУВАННЯ

Візьмемо за умову вже готові рішення площ трикутників за координатами їх вершин:

- **1 трикутник**

2 способ:

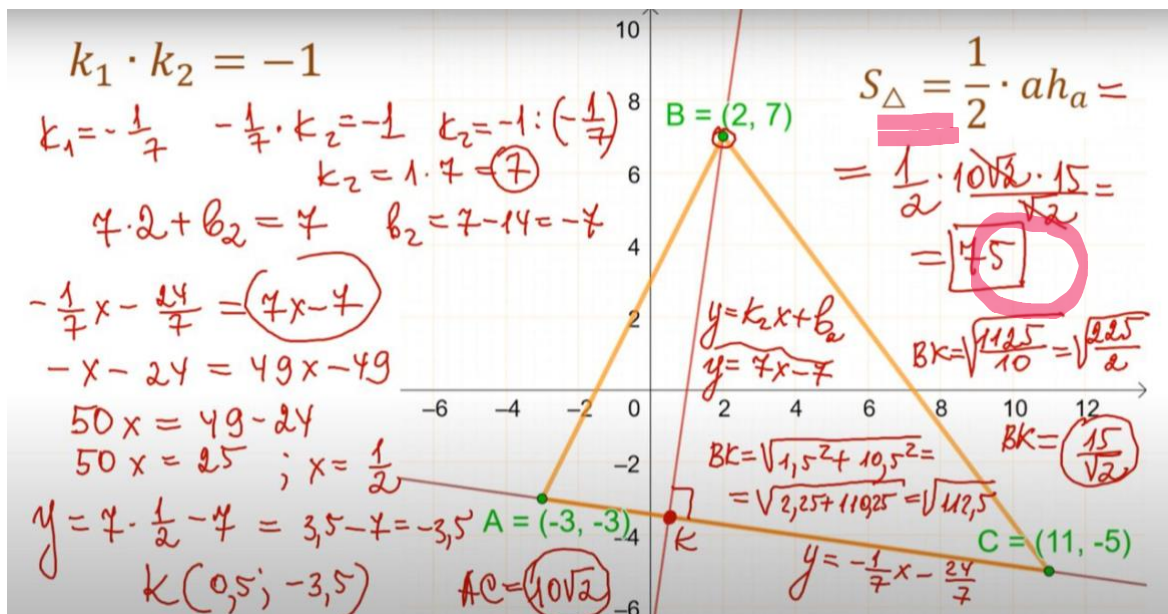
Подставляем в формулу $x_1=-1, y_1=-3, x_2=3, y_2=4, x_3=5, y_3=-5$:

$$\begin{aligned} S_{\Delta ABC} &= \frac{1}{2} |(3 - (-1))(-5 - (-3)) - (5 - (-1))(4 - (-3))| = \\ &= \frac{1}{2} |4 \cdot (-2) - 6 \cdot 7| = \frac{1}{2} \cdot 50 = \underline{25}. \end{aligned}$$

- 2 трикутник

Обчисліть площу трикутника ABC за допомогою складання рівнянь прямих, вершини якого мають координати

$A(-3; -3), B(2; 7), C(11; -5)$.



Проведемо тепер тестування програмою:

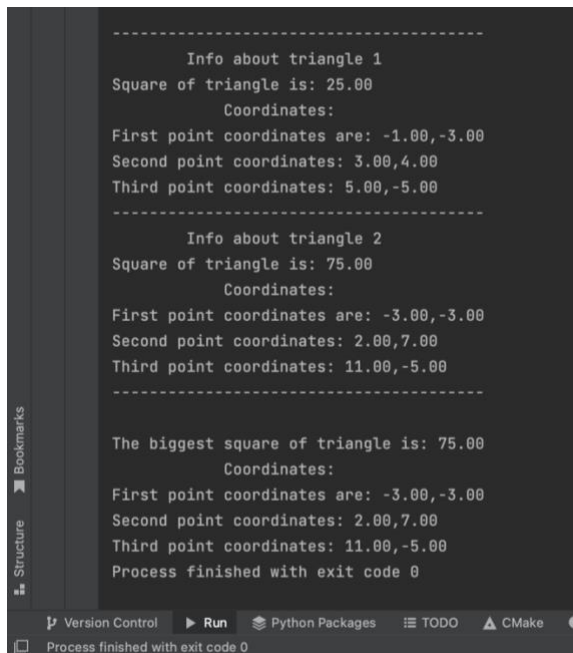
Ввід вхідних даних:

```

lab3 Point.h
Proj...
Run: lab3 x
/Users/mac/CLionProjects/la
Enter quantity of triangles
Enter triangle #1
Enter 1st point x: -3
Enter 1st point y: -3
Enter 2nd point x: 2
Enter 2nd point y: 7
Enter 3d point x: 11
Enter 3d point y: -5
Enter triangle #2
Enter 1st point x: -3
Enter 1st point y: -3
Enter 2nd point x: 2
Enter 2nd point y: 7
Enter 3d point x: 11
Enter 3d point y: -5

```

Вивід проміжних та кінцевих даних:



```
-----  
Info about triangle 1  
Square of triangle is: 25.00  
Coordinates:  
First point coordinates are: -1.00,-3.00  
Second point coordinates: 3.00,4.00  
Third point coordinates: 5.00,-5.00  
-----  
Info about triangle 2  
Square of triangle is: 75.00  
Coordinates:  
First point coordinates are: -3.00,-3.00  
Second point coordinates: 2.00,7.00  
Third point coordinates: 11.00,-5.00  
-----  
The biggest square of triangle is: 75.00  
Coordinates:  
First point coordinates are: -3.00,-3.00  
Second point coordinates: 2.00,7.00  
Third point coordinates: 11.00,-5.00  
Process finished with exit code 0
```

Отож, кінцеві дані зійшлися, площі знайдені та порівнянні правильно, можна вважати, що програма працює коректно.

Лінк на репозиторій у GitHub:

<https://github.com/Uliana200407/CppProjects-.git>