

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни

«Основи програмування 2. Модульне програмування»

«Перевантаження операторів»

Варіант 27

Виконав студент ІП-15, Пономаренко Маргарита Альбертівна
(шифр, прізвище, ім'я, по батькові)

Перевірила Муха Ірина Павлівна
(прізвище, ім'я, по батькові)

Лабораторна робота 3

Перевантаження операторів

Індивідуальне завдання

Варіант 27

27. Визначити клас "Точка", який задається координатами точки на площині. Реалізувати для нього декілька конструкторів, геттери, метод визначення квадранта системи координат (його номера), в якому знаходиться дана точка. Перевантажити оператори: префіксний "++" – для збільшення x-координати точки на 1, постфіксний "++" – для збільшення y-координати точки на 1, "-" – для визначення відстані між двома точками. Створити три точки (T1, T2, T3), використовуючи різні конструктори. Інкрементувати x-координату точки T1 і y-координату точки T2. Визначити відстань між отриманими точками T1 і T2. З'ясувати, якому квадранту належить точка T3.

Код

Point.h :

Основи програмування 2. Модульне програмування

```
Point.h  x  OP-Lab4(2 term).cpp
OP-Lab4(2 term)  (Глобальная область)

1  #pragma once
2  #include <math.h>
3  using namespace std;
4
5  class Point {
6  private:
7      int x;
8      int y;
9
10 public:
11     Point() { //конструктор за замовчуванням
12         x = 0;
13         y = 0;
14     }
15
16     Point(int xCoordinate, int yCoordinate) { //конструктор з параметрами
17         x = xCoordinate;
18         y = yCoordinate;
19     }
20
21     int GetXPoint() { return x; }
22     int GetYPoint() { return y; }
23
```

```
Point.h  x  OP-Lab4(2 term).cpp
OP-Lab4(2 term)  (Глобальная область)

22     int GetYPoint() { return y; }
23
24     int FindQuadrant() { //метод для визначення квадранта точки
25         if (x >= 0 && y >= 0) {
26             return 1;
27         }
28         if (x < 0 && y >= 0) {
29             return 2;
30         }
31         if (x < 0 && y < 0) {
32             return 3;
33         }
34         if (x >= 0 && y < 0) {
35             return 4;
36         }
37     }
38
39     Point& operator++() { //перевантаження префіксного "++" для збільшення x-координати точки на 1
40         ++x;
41         return *this; //розіменування покажчика
42     }
43
```

Основи програмування 2. Модульне програмування

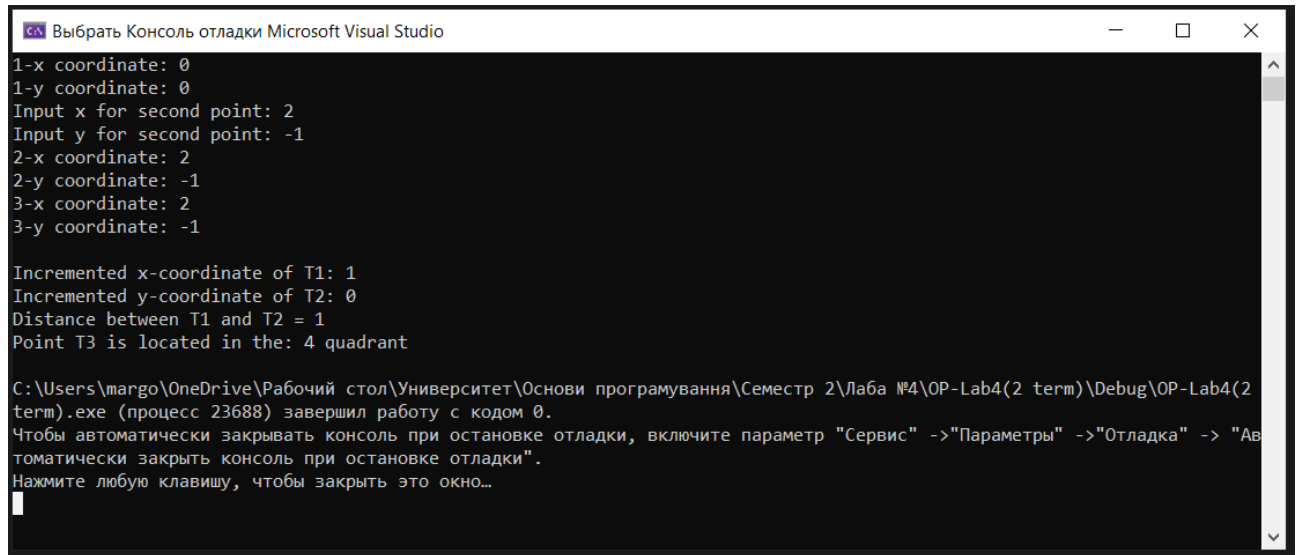
```
Point.h  OP-Lab4(2 term).cpp
OP-Lab4(2 term)  (Глобальная область)
34     if (x >= 0 && y < 0) {
35         return 4;
36     }
37 }
38
39 Point& operator++() { //перевантаження префіксного "++" для збільшення x-координати точки на 1
40     ++x;
41     return *this; //розіменування покажчика
42 }
43
44 Point& operator++(int coordinate) { //перевантаження постфіксного "++" для збільшення y-координати точки на 1
45     ++y;
46     return *this;
47 }
48
49 double operator-(const Point& other) { //перевантаження оператора "-" для визначення відстані між двома точками
50     double distance = sqrt(pow((other.x - this->x), 2) + pow((other.y - this->y), 2));
51     return distance;
52 }
53 };
```

OP-Lab4(2 term).cpp :

```
Point.h  OP-Lab4(2 term).cpp  main()
OP-Lab4(2 term)  (Глобальная область)
1  //Визначити клас "Точка", який задається координатами точки на площині. Реалізувати для нього декілька конструкторів, геттери,
2  //метод визначення квадранта системи координат (його номера), в якому знаходиться дана точка. Перевантажити оператори: префіксний "++" -
3  //для збільшення x-координати точки на 1, постфіксний "++" - для збільшення y-координати точки на 1, "-" - для визначення відстані
4  //між двома точками. Створити три точки (T1, T2, T3), використовуючи різні конструктори. Інкrementувати x-координату точки T1 і
5  //y-координату точки T2. Визначити відстань між отриманими точками T1 і T2. З'ясувати, якому квадранту належить точка T3.
6
7  #include <iostream>
8  #include "Point.h"
9  using namespace std;
10
11 int main()
12 {
13     Point first; //створення першої точки T1
14     cout << "1-x coordinate: " << first.GetXPoint() << endl << "1-y coordinate: " << first.GetYPoint() << endl;
15
16     int a;
17     int b;
18     cout << "Input x for second point: ";
19     cin >> a;
20     cout << "Input y for second point: ";
21     cin >> b;
22     Point second(a, b); //створення другої точки T2
23     cout << "2-x coordinate: " << second.GetXPoint() << endl << "2-y coordinate: " << second.GetYPoint() << endl;
```

```
Point.h  OP-Lab4(2 term).cpp
OP-Lab4(2 term)  (Глобальная область)
22     Point second(a, b); //створення другої точки T2
23     cout << "2-x coordinate: " << second.GetXPoint() << endl << "2-y coordinate: " << second.GetYPoint() << endl;
24
25     Point third = second; //копіювальна ініціалізація, створення точки T3
26     cout << "3-x coordinate: " << third.GetXPoint() << endl << "3-y coordinate: " << third.GetYPoint() << endl;
27
28     ++first;
29     cout << endl << "Incremented x-coordinate of T1: " << first.GetXPoint();
30
31     second++;
32     cout << endl << "Incremented y-coordinate of T2: " << second.GetYPoint();
33
34     double distanceBetween = first - second;
35     cout << endl << "Distance between T1 and T2 = " << distanceBetween;
36
37     cout << endl << "Point T3 is located in the: " << third.FindQuadrant() << " quadrant" << endl;
38
39 }
40
```

Результат виконання



The screenshot shows a 'Выбрать Консоль отладки Microsoft Visual Studio' window. The console output is as follows:

```
1-x coordinate: 0
1-y coordinate: 0
Input x for second point: 2
Input y for second point: -1
2-x coordinate: 2
2-y coordinate: -1
3-x coordinate: 2
3-y coordinate: -1

Incremented x-coordinate of T1: 1
Incremented y-coordinate of T2: 0
Distance between T1 and T2 = 1
Point T3 is located in the: 4 quadrant

C:\Users\margo\OneDrive\Рабочий стол\Университет\Основи програмування\Семестр 2\Лаба №4\0P-Lab4(2 term)\Debug\0P-Lab4(2 term).exe (процесс 23688) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```