

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 №4 з дисципліни  
«Основи програмування 2.  
Модульне програмування»

«Перевантаження операторів»

Варіант 32

Виконав студент ІІ-13 Шевцова Анастасія Андріївна  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №4  
Перевантаження операторів  
Варіант 32

Визначити клас «Трикутник», членами якого є координати вершин трикутника в просторі. Реалізувати для нього декілька конструкторів, геттери, методи обчислення периметра трикутника. Перевантажити оператори: префіксний «++» / постфіксний «++» - для інкрементування усіх x-координат і усіх y-координат вершин трикутника відповідно, «+=» - для збільшення усіх координат вершин трикутника на вказану величину. Створити три трикутника (T1, T2, T3), використовуючи різні конструктори. Інкрементувати x-координати вершин трикутника T1 і y-координати вершин трикутника T2. Збільшити координати вершин трикутника T3 на вказану величину. Серед трикутників T1, T2, T3 визначити трикутник, що має найбільший периметр.

C++

lab\_4.cpp

```
#include "triangle.h"
#include "point.h"

int main()
{
    triangle T1;

    std::cout << "Enter the vertices of triangle T2." << std::endl;
    std::cout << "Enter first vertex:" << std::endl;
    point vertex1; vertex1.inputPoint();
    std::cout << "Enter second vertex:" << std::endl;
    point vertex2; vertex2.inputPoint();
    std::cout << "Enter third vertex:" << std::endl;
    point vertex3; vertex3.inputPoint();
    triangle T2(vertex1, vertex2, vertex3);

    std::cout << "Enter first vertex of triangle T3:" << std::endl;
    point vertexT3; vertexT3.inputPoint();
    triangle T3(vertexT3);

    std::cout << "number = ";
    double number; std::cin >> number;

    std::cout << "\n-----Original triangles-----" << std::endl;
    std::cout << "T1: " << std::endl; outputTriangle(T1);
    std::cout << "T2: " << std::endl; outputTriangle(T2);
    std::cout << "T3: " << std::endl; outputTriangle(T3);
    std::cout << "-----" << std::endl;

    ++T1;
    T2++;
    T3 += number;

    std::cout << "\n-----Changed triangles-----" << std::endl;
    std::cout << "T1: " << std::endl; outputTriangle(T1);
    std::cout << "T2: " << std::endl; outputTriangle(T2);
    std::cout << "T3: " << std::endl; outputTriangle(T3);
    std::cout << "-----" << std::endl;

    largestPerimetr(T1, T2, T3);

    return 0;
}
```

### triangle.cpp

```
#include "triangle.h"
#include "point.h"

//class constructors

triangle::triangle()
{
    firstVertex.setX(0);
    firstVertex.setY(0);
    firstVertex.setZ(0);

    secondVertex.setX(1);
    secondVertex.setY(1);
    secondVertex.setZ(1);

    thirdVertex.setX(2);
    thirdVertex.setY(2);
    thirdVertex.setZ(2);
}

triangle::triangle(point firstVertex, point secondVertex, point thirdVertex)
{
    this->firstVertex = firstVertex;
    this->secondVertex = secondVertex;
    this->thirdVertex = thirdVertex;
}

triangle::triangle(const triangle &T)
{
    this->firstVertex = T.firstVertex;
    this->secondVertex = T.secondVertex;
    this->thirdVertex = T.thirdVertex;
}

triangle::triangle(point firstVertex1) : firstVertex(firstVertex1), secondVertex(0,
1, 2), thirdVertex(1, 2, 3) {}

//getters

point triangle::getFirstVertex() { return firstVertex; }

point triangle::getSecondVertex() { return secondVertex; }

point triangle::getThirdVertex() { return thirdVertex; }

//operator overload

triangle& triangle::operator++()
{
    this->firstVertex.setX(this->firstVertex.getX() + 1);
    this->secondVertex.setX(this->secondVertex.getX() + 1);
    this->thirdVertex.setX(this->thirdVertex.getX() + 1);
    return *this;
}

triangle& triangle::operator++(int a)
{
    this->firstVertex.setY(this->firstVertex.getY() + 1);
    this->secondVertex.setY(this->secondVertex.getY() + 1);
    this->thirdVertex.setY(this->thirdVertex.getY() + 1);
    return *this;
}
```

```

triangle& triangle::operator+=(double value)
{
    this->firstVertex.setX(this->firstVertex.getX() + value);
    this->firstVertex.setY(this->firstVertex.getY() + value);
    this->firstVertex.setZ(this->firstVertex.getZ() + value);

    this->secondVertex.setX(this->firstVertex.getX() + value);
    this->secondVertex.setY(this->firstVertex.getY() + value);
    this->secondVertex.setZ(this->firstVertex.getZ() + value);

    this->thirdVertex.setX(this->firstVertex.getX() + value);
    this->thirdVertex.setY(this->firstVertex.getY() + value);
    this->thirdVertex.setZ(this->firstVertex.getZ() + value);

    return *this;
}

//other class methods

double triangle::perimetr()
{
    double x1 = firstVertex.getX();
    double y1 = firstVertex.getY();
    double z1 = firstVertex.getZ();

    double x2 = secondVertex.getX();
    double y2 = secondVertex.getY();
    double z2 = secondVertex.getZ();

    double x3 = thirdVertex.getX();
    double y3 = thirdVertex.getY();
    double z3 = thirdVertex.getZ();

    double side1 = sqrt(pow((x1 - x2), 2) + pow((y1 - y2), 2) + pow((z1 - z2), 2));
    double side2 = sqrt(pow((x2 - x3), 2) + pow((y2 - y3), 2) + pow((z2 - z3), 2));
    double side3 = sqrt(pow((x3 - x1), 2) + pow((y3 - y1), 2) + pow((z3 - z1), 2));

    double P = side1 + side2 + side3;
    return P;
}

//functions

void largestPerimetr(triangle T1, triangle T2, triangle T3)
{
    double P1 = T1.perimetr();
    double P2 = T2.perimetr();
    double P3 = T3.perimetr();

    if (P1 > P2) {
        if (P1 > P3) {
            std::cout << "T1 has the largest perimetr: " << P1 << std::endl;
        }
        else {
            std::cout << "T3 has the largest perimetr: " << P3 << std::endl;
        }
    }
    else if (P2 > P3) {
        std::cout << "T2 has the largest perimetr: " << P2 << std::endl;
    }
    else {
        std::cout << "T3 has the largest perimetr: " << P3 << std::endl;
    }
}

void outputTriangle(triangle Triangle)
{
    point firstVertex = Triangle.getFirstVertex();
    point secondVertex = Triangle.getSecondVertex();

```

```
point thirdVertex = Triangle.getThirdVertex();

std::cout << "First vertex:" << std::endl;
firstVertex.printPoint();
std::cout << "Second vertex:" << std::endl;
secondVertex.printPoint();
std::cout << "Third vertex:" << std::endl;
thirdVertex.printPoint();
}
```

### triangle.h

```
#pragma once
#include "point.h"

class triangle
{
private:
    point firstVertex;
    point secondVertex;
    point thirdVertex;
public:
    triangle();
    triangle(point, point, point);
    triangle(const triangle&);
    triangle(point);

    point getFirstVertex();
    point getSecondVertex();
    point getThirdVertex();

    triangle& operator ++();
    triangle& operator ++(int);
    triangle& operator +=(double);

    double perimetr();
};

void largestPerimetr(triangle, triangle, triangle);
void outputTriangle(triangle);
```

### point.cpp

```
#include "point.h"

//class constructors

point::point()
{
    x = 0;
    y = 0;
    z = 0;
}

point::point(double x, double y, double z)
{
    this->x = x;
    this->y = y;
    this->z = z;
}

point::point(const point& Point)
{
    this->x = Point.x;
    this->y = Point.y;
    this->z = Point.z;
}
```

```
//setters
void point::setX(double num) { x = num; }
void point::setY(double num) { y = num; }
void point::setZ(double num) { z = num; }

//getters
double point::getX() { return x; }
double point::getY() { return y; }
double point::getZ() { return z; }

//other class methods
void point::printPoint()
{
    std::cout << "x = " << x << "\ty = " << y << "\tz = " << z << std::endl;
}

void point::inputPoint()
{
    std::cout << "x = "; std::cin >> x;
    std::cout << "y = "; std::cin >> y;
    std::cout << "z = "; std::cin >> z;
}
```

point.h

```
#pragma once
#include <iostream>
#include <cmath>

class point
{
private:
    double x;
    double y;
    double z;
public:
    point();
    point(double, double, double);
    point(const point&);

    void setX(double);
    void setY(double);
    void setZ(double);

    double getX();
    double getY();
    double getZ();

    void printPoint();
    void inputPoint();
};
```

## Тестування:

```
Enter the vertices of triangle T2.
Enter first vertex:
x = 2
y = 5
z = 1
Enter second vertex:
x = 1
y = 1
z = 1
Enter third vertex:
x = 2
y = -6
z = 2
Enter first vertex of triangle T3:
x = 3
y = 4
z = 0
number = 3
```

-----Original triangles-----

T1:

First vertex:

x = 0 y = 0 z = 0

Second vertex:

x = 1 y = 1 z = 1

Third vertex:

x = 2 y = 2 z = 2

T2:

First vertex:

x = 2 y = 5 z = 1

Second vertex:

x = 1 y = 1 z = 1

Third vertex:

x = 2 y = -6 z = 2

T3:

First vertex:

x = 3 y = 4 z = 0

Second vertex:

x = 0 y = 1 z = 2

Third vertex:

x = 1 y = 2 z = 3

-----

-----Changed triangles-----

T1:

First vertex:

x = 1 y = 0 z = 0

Second vertex:

x = 2 y = 1 z = 1

Third vertex:

x = 3 y = 2 z = 2

T2:

First vertex:

x = 2 y = 6 z = 1

Second vertex:

x = 1 y = 2 z = 1

Third vertex:

x = 2 y = -5 z = 2

T3:

First vertex:

x = 6 y = 7 z = 3

Second vertex:

x = 3 y = 4 z = 5

Third vertex:

x = 4 y = 5 z = 6

-----

T2 has the largest perimetr: 22.3099