

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 ПІ-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"
```

```
int main()
{
    triangle T1;

    cout << "Enter the vertices of triangle T2." << endl;
    cout << "Enter first vertex:" << endl;
    vector<double> vertex1 = inputVertex();
    cout << "Enter second vertex:" << endl;
    vector<double> vertex2 = inputVertex();
    cout << "Enter third vertex:" << endl;
    vector<double> vertex3 = inputVertex();
    triangle T2(vertex1, vertex2, vertex3);

    cout << "Enter first vertex of triangle T3:" << endl;
    vector<double> vertexT3 = inputVertex();
    triangle T3(vertexT3);

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

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

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

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

    largestPerimetr(T1, T2, T3);

    return 0;
}
```

triangle.cpp

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

```
//class constructors
```

```
triangle::triangle()
```

```
{  
    firstVertex = { 0, 0, 0 };  
    secondVertex = { 1, 1, 1 };  
    thirdVertex = { 2, 2, 2 };  
}
```

```
triangle::triangle(vector<double>& firstVertex, vector<double>& secondVertex,  
vector<double>& 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(vector<double>& firstVertex)
```

```
{  
    this->firstVertex = firstVertex;  
    secondVertex = { 0, 1, 2 };  
    thirdVertex = { 1, 2, 3 };  
}
```

```
//getters
```

```
vector<double> triangle::getFirstVertex()
```

```
{  
    return firstVertex;  
}
```

```
vector<double> triangle::getSecondVertex()
```

```
{  
    return secondVertex;  
}
```

```
vector<double> triangle::getThirdVertex()
```

```
{  
    return thirdVertex;  
}
```

```
//operator overload
```

```
triangle& triangle::operator++()
```

```
{  
    this->firstVertex[0]++;  
    this->secondVertex[0]++;  
    this->thirdVertex[0]++;  
    return *this;  
}
```

```
triangle& triangle::operator++(int a)
```

```
{  
    this->firstVertex[1]++;  
    this->secondVertex[1]++;  
    this->thirdVertex[1]++;  
    return *this;  
}
```

```

triangle& triangle::operator+=(double number)
{
    for (int i = 0; i < firstVertex.size(); i++)
    {
        firstVertex[i] += number;
    }
    for (int i = 0; i < secondVertex.size(); i++)
    {
        secondVertex[i] += number;
    }
    for (int i = 0; i < thirdVertex.size(); i++)
    {
        thirdVertex[i] += number;
    }
    return *this;
}

//other class methods

double triangle::perimetr()
{
    double x1 = firstVertex[0];
    double y1 = firstVertex[1];
    double z1 = firstVertex[2];

    double x2 = secondVertex[0];
    double y2 = secondVertex[1];
    double z2 = secondVertex[2];

    double x3 = thirdVertex[0];
    double y3 = thirdVertex[1];
    double z3 = thirdVertex[2];

    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;
}

void triangle::outputTriangle()
{
    cout << "First vertex:" << endl;
    cout << "x = " << firstVertex[0] << "\ty = " << firstVertex[1] << "\tz = " <<
firstVertex[2] << endl;
    cout << "Second vertex:" << endl;
    cout << "x = " << secondVertex[0] << "\ty = " << secondVertex[1] << "\tz = " <<
secondVertex[2] << endl;
    cout << "Third vertex:" << endl;
    cout << "x = " << thirdVertex[0] << "\ty = " << thirdVertex[1] << "\tz = " <<
thirdVertex[2] << endl;
}

//functions

vector<double> inputVertex()
{
    vector<double> vertex(3);
    cout << "x = "; cin >> vertex[0];
    cout << "y = "; cin >> vertex[1];
    cout << "z = "; cin >> vertex[2];
    return vertex;
}

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) {
        cout << "T1 has the largest perimetr: " << P1 << endl;
    }
    else {
        cout << "T3 has the largest perimetr: " << P3 << endl;
    }
}
else if (P2 > P3) {
    cout << "T2 has the largest perimetr: " << P2 << endl;
}
else {
    cout << "T31 has the largest perimetr: " << P3 << endl;
}
}

triangle.h
#pragma once
#include <iostream>
#include <vector>
#include <cmath>
using namespace std;

class triangle
{
private:
    vector<double> firstVertex;
    vector<double> secondVertex;
    vector<double> thirdVertex;
public:
    triangle();
    triangle(vector<double>&, vector<double>&, vector<double>&);
    triangle(const triangle&);
    triangle(vector<double>&);

    vector<double> getFirstVertex();
    vector<double> getSecondVertex();
    vector<double> getThirdVertex();

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

    double perimetr();
    void outputTriangle();
};

vector<double> inputVertex();
void largestPerimetr(triangle, triangle, triangle);
```

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
```