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.
Методології програмування»

«Перевантаження операторів»

Варіант 20

Виконав студент ІП-15, Ликова Катерина Олександрівна
(шифр, прізвище, ім’я, по батькові)

Перевірила Вєчерковська Анастасія Сергіївна
(прізвище, ім’я, по батькові)

Київ 2022

Мета роботи: вивчити механізми створення класів з використанням перевантажених операторів (операцій).

Постановка задачі:

20. Визначити клас "Трикутник", членами якого є сторони трикутника в просторі. Реалізувати для нього декілька конструкторів, геттери, методи обчислення площі трикутника. Перевантажити оператори: "++" / "--" - для інкрементування / декрементування довжин сторін трикутника відповідно, "+=" / "-=" – для збільшення / зменшення довжин сторін трикутника на вказану величину. Створити три трикутника (T1, T2, T3), використовуючи різні конструктори. Інкрементувати довжини сторін трикутника T1, а довжини сторін трикутника T2 декрементувати. Збільшити довжини сторін трикутника T3 на вказану величину. Серед трикутників T1, T2, T3 визначити трикутник, що має найбільшу площу.

Код

C++

main.cpp

```
#include "func.h"
int main()
{
	srand(time(NULL));
	cout << "Triangle 1 is created automaticaly" << endl << endl;
	Triangle trian1;
	cout << "Triangle 2:" << endl;
	double side1 = set_side(1);
	Triangle trian2(side1);
	cout << endl << "Triangle 3:" << endl;
	side1 = set_side(1);
	double side2 = set_side(2);
	double side3 = set_side(3);
	Triangle trian3(side1, side2, side3);
	cout << endl << "Triangle 1:" << endl;
	trian1.print_sides();
	cout << "Triangle 2:" << endl;
	trian2.print_sides();
	cout << "Triangle 3:" << endl;
	trian3.print_sides();
	++trian1;
	cout << "Triangle 1 after incrimination of sides:" << endl;
	trian1.print_sides();
	--trian2;
	cout << "Triangle 2 after decrementation of sides:" << endl;
	trian2.print_sides();
	double num;
	cout << "Enter enter the number by which you want to raise the sides of the triangle 3: ";
	num = set_num(trian3.get_side1(), trian3.get_side2(), trian3.get_side3());
	trian3 += num;
```

```
	cout << "Triangle 3 after adding a number to sides:" << endl;
	trian3.print_sides();
	Triangle trian_big = find_biggest_area(trian1, trian2, trian3);
	cout << endl << "Triangle with biggest area:" << endl;
	trian_big.print_sides();
	return 0;
}
```

func.h

```
#pragma once
#include <string>
#include <iostream>
#include <time.h>
#include <math.h>
#include "Triangle.h"
using namespace std;
double set_side(int);
double set_num(double, double, double);
Triangle find_biggest_area(Triangle, Triangle, Triangle);
```

func.cpp

```
#include "func.h"
double set_side(int num)
{
	cout << "Input side " << num << ": ";
	string side;
	getline(cin, side);
	int i = 0;
	while (i < size(side) || size(side) == 0)
	{
		if ((!isdigit(side[i]) && side[i] != '.') || side[0] == '0')
		{
			cout << "Length of side is incorrect. Please enter correct length: ";
			getline(cin, side);
			i = 0;
		}
		else
		{
			i++;
		}
	}
	if (stod(side) < 1.1 || stod(side) >= 100)
	{
		cout << "Side length is incorrect. Enter length in the interval (1.1 <= length < 100): " <<
endl;
		set_side(num);
	}
	return stod(side);
}
double set_num(double side1, double side2, double side3)
{
	string num;
	getline(cin, num);
	int i = 0;
```

```
	while (i < size(num) || size(num) == 0)
	{
		if ((!isdigit(num[i]) && num[i] != '.') || num[0] == '0')
		{
			cout << "Number is incorrect. Please enter correct number: ";
			getline(cin, num);
			i = 0;
		}
		else
		{
			i++;
		}
	}
	return stod(num);
}
Triangle find_biggest_area(Triangle trian1, Triangle trian2, Triangle trian3)
{
	double s1 = trian1.area();
	double s2 = trian2.area();
	double s3 = trian3.area();
	cout << "Area of triangle 1: " << s1 << endl;
	cout << "Area of triangle 2: " << s2 << endl;
	cout << "Area of triangle 3: " << s3 << endl;
	double max = s1;
	if (s1 == s2 && s2 == s3)
	{
		cout << "Triangles 1, 2 and 3 have the same area: " << max << endl;
		return trian1;
	}
	else if (s1 == s2)
	{
		if (max < s3)
		{
			max = s3;
			cout << "Triangle 3 has biggest area: " << max << endl;
			return trian3;
		}
		cout << "Triangles 1 and 2 have biggest area: " << max << endl;
		return trian1;
	}
	else if (s1 == s3)
	{
		if (max < s2)
		{
			max = s2;
			cout << "Triangle 2 has biggest area: " << max << endl;
			return trian2;
		}
		cout << "Triangles 1 and 3 have biggest area: " << max << endl;
		return trian1;
	}
	else if (s2 == s3)
```

```
	{
		if (max < s2)
		{
			max = s2;
			cout << "Triangles 2 and 3 have biggest area: " << max << endl;
			return trian2;
		}
		cout << "Triangle 1 has biggest area: " << max << endl;
		return trian1;
	}
	else
	{
		if (max < s2)
		{
			max = s2;
			if (max < s3)
			{
				max = s3;
				cout << "Triangle 3 has biggest area: " << max << endl;
				return trian3;
			}
			cout << "Triangle 2 has biggest area: " << max << endl;
			return trian2;
		}
		else if (max < s3)
		{
			max = s3;
			cout << "Triangle 3 has biggest area: " << max << endl;
			return trian3;
		}
		cout << "Triangle 1 has biggest area: " << max << endl;
		return trian1;
	}
}
```

Triangle_arr.h

```
#pragma once
class Triangle
{
public:
	Triangle();
	Triangle(double);
	Triangle(double, double, double);
	void print_sides();
	double get_side1();
	double get_side2();
	double get_side3();
	double area();
	Triangle& operator++();
	Triangle& operator--();
	Triangle& operator+=(double);
	Triangle& operator-=(double);
private:
```

```
        double side1;
        double side2;
        double side3;
};
```

Triangle.cpp

```
#include "func.h"
Triangle::Triangle()
{
        side1 = 1 + 0.1* (rand() % 980);
        side2 = 1 + 0.1 * (rand() % 980);
        side3 = 1 + 0.1 * (rand() % 980);
        while ((side2 >= (side1 + side3)) || (side1 >= (side2 + side3)) || (side3 >= (side1 + side2)))
        {
                side3 = 1 + 0.1 * (rand() % 980);
        }
}
void Triangle::print_sides()
{
        cout << "Side 1: " << side1 << endl;
        cout << "Side 2: " << side2 << endl;
        cout << "Side 3: " << side3 << endl << endl;
}
Triangle::Triangle(double side11)
{
        side1 = side11;
        side2 = 1 + 0.1 * (rand() % 980);
        side3 = 1 + 0.1 * (rand() % 980);
        while ((side2 >= (side1 + side3)) || (side1 >= (side2 + side3)) || (side3 >= (side1 + side2)))
        {
                side3 = 1 + 0.1 * (rand() % 980);
        }
}
Triangle::Triangle(double side11, double side22, double side33)
{
        side1 = side11;
        side2 = side22;
        side3 = side33;
        while ((side2 >= (side1 + side3)) || (side1 >= (side2 + side3)) || (side3 >= (side1 + side2)))
        {
                cout << "Length of side 3 is incorrect. ";
                side3 = set_side(3);
        }
}
double Triangle::get_side1()
{
        return side1;
}
double Triangle::get_side2()
{
        return side2;
}
double Triangle::get_side3()
```

```
{
	return side3;
}
double Triangle::area()
{
	double area;
	if ((side2 >= (side1 + side3)) || (side1 >= (side2 + side3)) || (side3 >= (side1 + side2)))
	{
		return 0;
	}
	else
	{
		double half_perimeter = (side1 + side2 + side3) / 2;
		area = pow(half_perimeter * (half_perimeter - side1) * (half_perimeter - side2) *
(half_perimeter - side3), 0.5);
		return area;
	}
}
Triangle& Triangle::operator++()
{
	++side1;
	++side2;
	++side3;
	return *this;
}
Triangle& Triangle::operator--()
{
	--side1;
	--side2;
	--side3;
	return *this;
}
Triangle& Triangle::operator+=(double num)
{
	side1 += num;
	side2 += num;
	side3 += num;
	return *this;
}
Triangle& Triangle::operator-=(double num)
{
	side1 -= num;
	side2 -= num;
	side3 -= num;
	return *this;
}
```

Тестування

C++

Microsoft Visual Studio Debug Console

```
Triangle 1 is created automaticaly

Triangle 2:
Input side 1: 34.8

Triangle 3:
Input side 1: 235
Side length is incorrect. Enter length in the interval (1.1 <= length < 100):
Input side 1: 45.8
Input side 2: 62
Input side 3: 34.5

Triangle 1:
Side 1: 72.3
Side 2: 64.8
Side 3: 67.7

Triangle 2:
Side 1: 34.8
Side 2: 21.9
Side 3: 20.1

Triangle 3:
Side 1: 45.8
Side 2: 62
Side 3: 34.5

Triangle 1 after incrimination of sides:
Side 1: 73.3
Side 2: 65.8
Side 3: 68.7

Triangle 2 after decrementation of sides:
Side 1: 33.8
Side 2: 20.9
Side 3: 19.1

Enter the number by which you want to raise the sides of the triangle 3: 44.5
Triangle 3 after adding a number to sides:
Side 1: 90.3
Side 2: 106.5
Side 3: 79

Area of triangle 1: 2064.92
Area of triangle 2: 180.495
Area of triangle 3: 3484.24
Triangle 3 has biggest area: 3484.24

Triangle with biggest area:
Side 1: 90.3
Side 2: 106.5
Side 3: 79
```