



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**« МИРЭА Российский технологический университет »**

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Вычислительной техники

**УЧЕБНОЕ ЗАДАНИЕ**

по дисциплине

« Объектно-ориентированное программирование »

Наименование задачи:

**« Задача 9\_1\_1 »**

С тудент группы

ИКБО-13-21

Черномуров С.А.

Руководитель практики

Ассистент

Асадова Ю.С.

Работа представлена

«\_\_»\_\_\_\_\_ 2022 г.

\_\_\_\_\_

(подпись студента)

Оценка

\_\_\_\_\_

(подпись руководителя)

Москва 2022

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Постановка задачи.....	5
Метод решения.....	7
Описание алгоритма.....	9
Блок-схема алгоритма.....	15
Код программы.....	22
Тестирование.....	25
ЗАКЛЮЧЕНИЕ.....	26
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	27

## **ВВЕДЕНИЕ**

## Постановка задачи

Перегрузка арифметических операций.

Перезагрузка операции для объекта треугольник.

У треугольника есть стороны  $a$ ,  $b$ ,  $c$  и они принимают только натуральные значения. Определяем операцию сложения и вычитания для треугольников.

$+$  сложить значения сторон, если допустимо.

$-$  вычесть значения сторон, если допустимо.

Складываются и вычитаются соответствующие стороны треугольников. Т.е.  $a_1 + a_2$ ,  $b_1 + b_2$ ,  $c_1 + c_2$ . Если после выполнения операции получается недопустимый треугольник, то результатом операции берется первый аргумент.

Написать программу, которая выполняет операции над треугольниками.

В основной программе реализовать алгоритм:

1. Ввод количества треугольников  $n$ .
2. В цикле для каждого треугольника вводятся исходные длины сторон. Далее создается объект, в конструктор которого передаются значения длин сторон. Каждый объект треугольника получает свой номер от 1 до  $n$ .
3. В цикле, последовательно, построчно вводится «номер первого треугольника» «символ арифметической операции  $+$  или  $-$ » «номер второго треугольника»
4. После каждого ввода выполняется операция, результат присваивается первому аргументу (объекту треугольника).
5. Цикл завершается по завершению данных.
6. Выводится результат последней операции.

Гарантируется:

- Количество треугольников больше или равно 2;
- Значения исходных длин сторон треугольников задаются корректно.

Реализовать перегрузку арифметических операции «+» и «-» для объектов треугольника посредством самостоятельных функций.

### **Описание входных данных**

Первая строка содержит значение количества треугольников  $n$ :  
«Натуральное значение»

Далее  $n$  строк содержат  
«Натуральное значение» «Натуральное значение»  
«Натуральное значение»

Начиная с  $n + 2$  строки:  
«Натуральное значение» «Знак операции» «Натуральное значение»

### **Описание выходных данных**

$a$  = «Натуральное значение»;  $b$  = «Натуральное значение»;  
 $c$  = «Натуральное значение».

## Метод решения

Для решения поставленной задачи используются:

- Объекты стандартных потоков ввода и вывода `cin` и `cout` соответственно. Используются для ввода с клавиатуры и вывода на экран.
- Объекты класса `Triangle` в количестве, определяемом пользователем.
- Функции-операторы "+" и "-". Используются для объектов класса `Triangle`.
- **Класс `Triangle`:**
  - Свойства/поля:
    - Поле:
      - Наименование - `a`, `b`, `c`;
      - Тип - целочисленный беззнаковый;
      - Модификатор доступа - закрытый.
  - Методы:
    - Метод `Triangle`:
      - Функционал - параметризированный конструктор.
    - Метод `Side1`:
      - Функционал - константный метод, возвращающий длину первой стороны треугольника.
    - Метод `Side2`:
      - Функционал - константный метод, возвращающий длину второй стороны треугольника.
    - Метод `Side3`:

- Функционал - константный метод, возвращающий длину третьей стороны треугольника.

## Описание алгоритма

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

Функция: main

Функционал: Основной алгоритм программы

Параметры: Отсутствуют

Возвращаемое значение: Целочисленный тип данных - код возврата

Алгоритм функции представлен в таблице 1.

Таблица 1. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной n	2	
2		Считывание с клавиатуры значения переменной n	3	
3		Создание объекта vec класса vector для хранения объектов класса Triangle	4	
4		Объявление целочисленной переменной c инициализацией i=0	5	Использование i в качестве счетчика
5	Значение i меньше значения n	Объявление целочисленных беззнаковых переменных a,b,c	6	
			10	Выход из цикла
6		Считывание с клавиатуры значений переменных a, b, c	7	
7		Создание объекта trg класса Triangle путем вызова	8	



		параметризированного конструктора с аргументами a, b, c		
8		Вызов метода push_back объекта vec с параметром trg	9	
9		Инкрементирование i	5	
10		Объявление целочисленных переменных num1, num2 и символьной переменной oper	11	
11	Значения num1, oper, num2 считаны с клавиатуры	Считывание с клавиатуры значений переменных num1, oper, num2	12	
			13	Выход из цикла
12	Значение oper == '+'	Присвоение значения num1+num2 значению num1	11	
		Присвоение значения num1-num2 значению num1	11	
13		Вывод на экран "a = ", ";", "b = ", ";", "c = ", соответствующие значения полей объекта trg	Ø	

Конструктор класса: Triangle

Модификатор доступа: public

Функционал: Параметризированный конструктор

Параметры: Целочисленные параметры A, B, C

Алгоритм конструктора представлен в таблице 2.

Таблица 2. Алгоритм конструктора класса Triangle

№	Предикат	Действия	№ перехода	Комментарий
1		Присвоение значений А, В, С соответствующим полям а, b, с объекта класса Triangle	Ø	

Класс объекта: Triangle

Модификатор доступа: public

Метод: Side1

Функционал: Константный метод, возвращающий длину первой стороны треугольника

Параметры: Отсутствуют

Возвращаемое значение: Целочисленный тип данных - длина первой стороны треугольника

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода Side1 класса Triangle

№	Предикат	Действия	№ перехода	Комментарий
1		Возврат методом значения поля а объекта класса Triangle	Ø	

Класс объекта: Triangle

Модификатор доступа: public

Метод: Side2

Функционал: Константный метод, возвращающий длину второй стороны треугольника

Параметры: Отсутствуют

Возвращаемое значение: Целочисленный тип данных - длина второй стороны треугольника

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода Side2 класса Triangle

№	Предикат	Действия	№ перехода	Комментарий
1		Возврат методом значения поля b объекта класса Triangle	Ø	

Класс объекта: Triangle

Модификатор доступа: public

Метод: Side3

Функционал: Константный метод, возвращающий длину третьей стороны треугольника

Параметры: Отсутствуют

Возвращаемое значение: Целочисленный тип данных - длина третьей стороны треугольника

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода Side3 класса Triangle

№	Предикат	Действия	№ перехода	Комментарий
1		Возврат методом значения поля c объекта класса Triangle	Ø	

Функция: operator+

Функционал: Суммирование длин сторон двух треугольников

Параметры: Ссылки на объекты trg1, trg2 класса Triangle

Возвращаемое значение: Объект класса Triangle

Алгоритм функции представлен в таблице 6.

Таблица 6. Алгоритм функции operator+

№	Предикат	Действия	№ перехода	Комментарий
1	Треугольник со сторонами А,В,С существует	Объявление целочисленных переменных А,В,С с инициализацией значениями сумм соответствующих полей объектов trg1, trg2	2	
		Возврат функцией объекта trg1	∅	
2		Возврат функцией значений А,В,С для созданного объекта класса Triangle	∅	

Функция: operator-

Функционал: Вычитание длин сторон двух треугольников

Параметры: Ссылки на объекты trg1, trg2 класса Triangle

Возвращаемое значение: Объект класса Triangle

Алгоритм функции представлен в таблице 7.

Таблица 7. Алгоритм функции operator-

№	Предикат	Действия	№ перехода	Комментарий
1	Треугольник со	Объявление	2	

	сторонами А,В,С существует	целочисленных переменных А,В,С с инициализацией значениями разностей соответствующих полей объектов trg1, trg2		
		Возврат функцией объекта trg1	∅	
2		Возврат функцией значений А,В,С для созданного объекта класса Triangle	∅	

## Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.



Рис. 1. Блок-схема алгоритма.

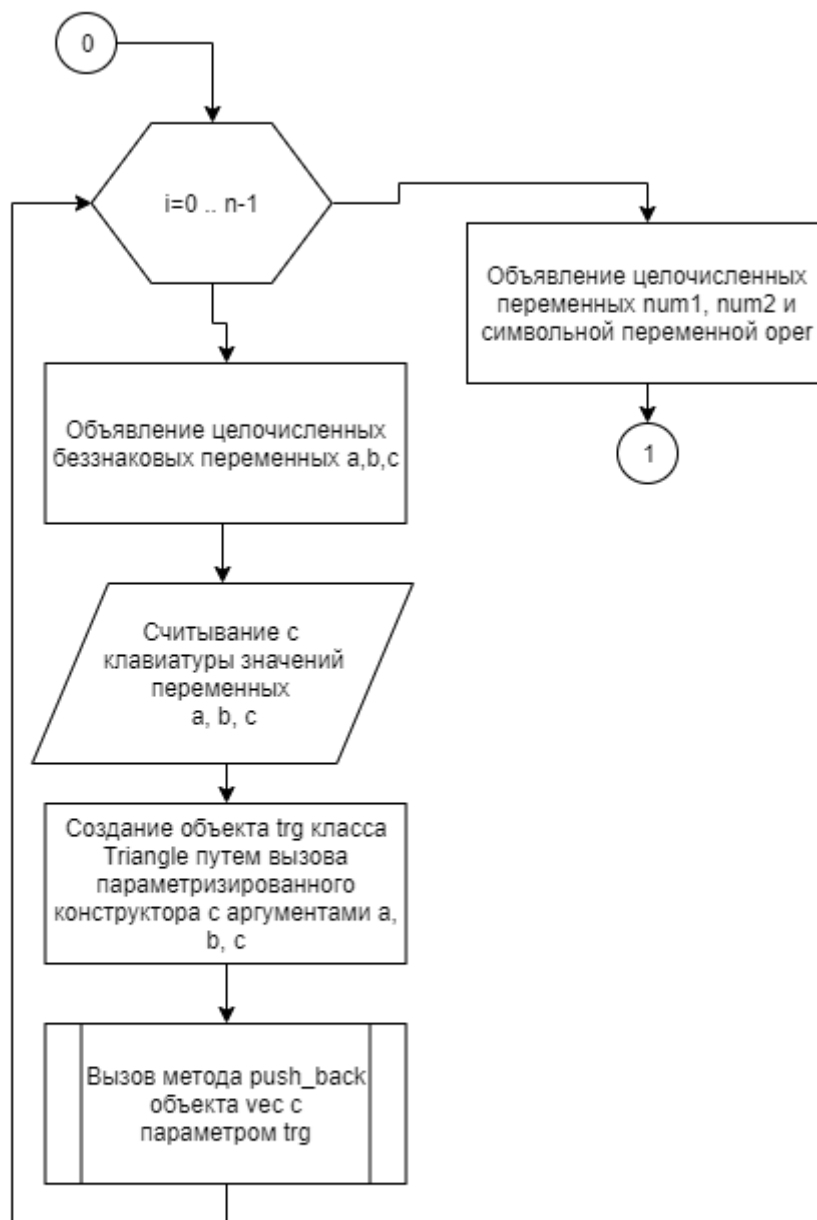


Рис. 2. Блок-схема алгоритма.

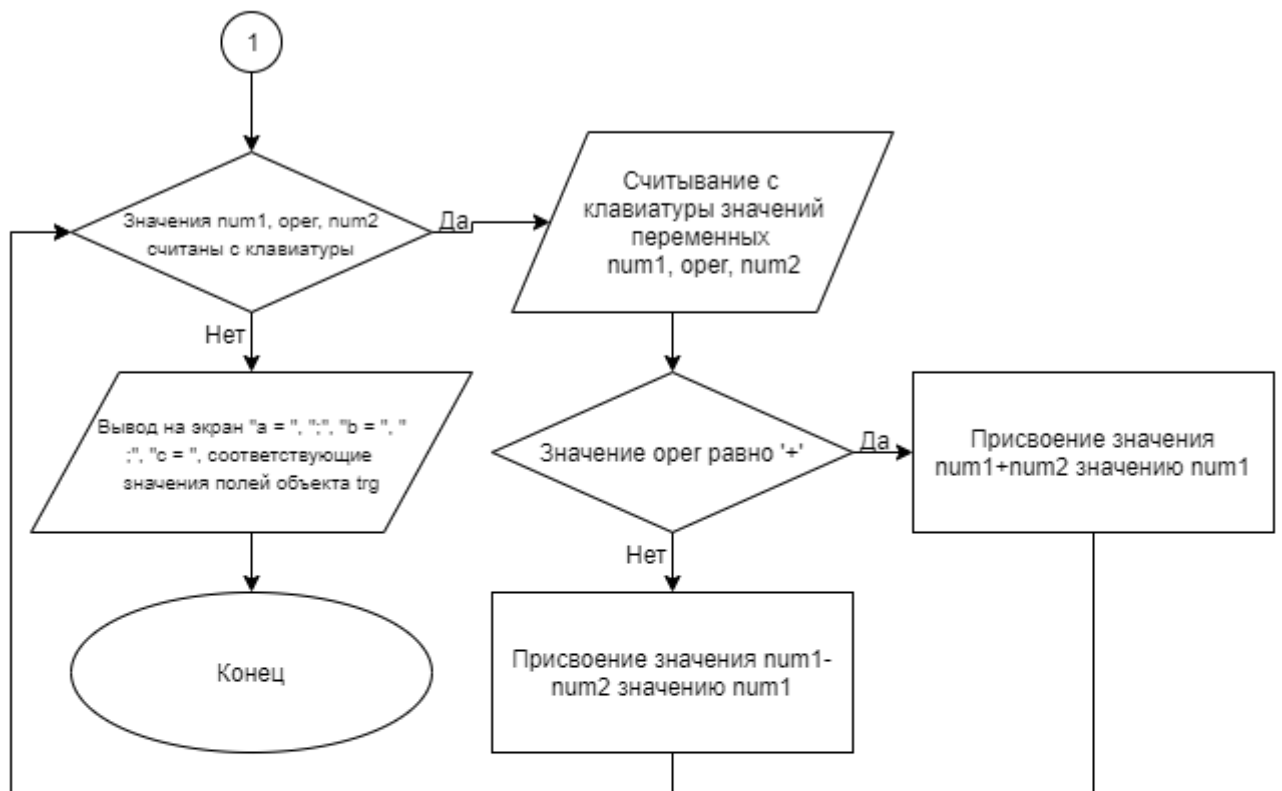


Рис. 3. Блок-схема алгоритма.



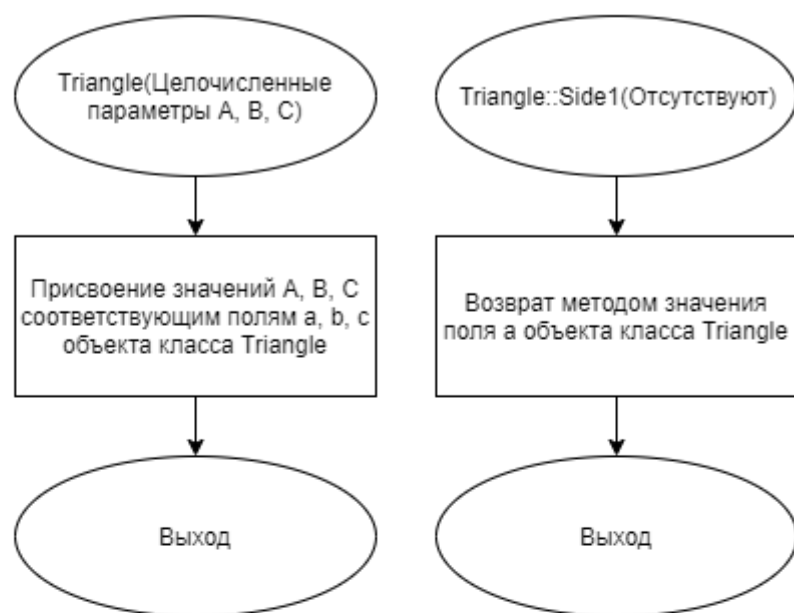


Рис. 4. Блок-схема алгоритма.

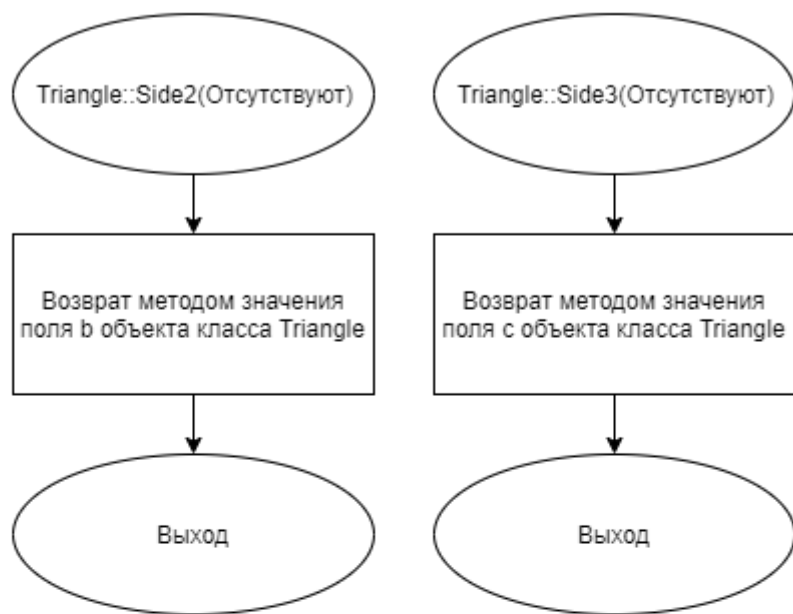


Рис. 5. Блок-схема алгоритма.

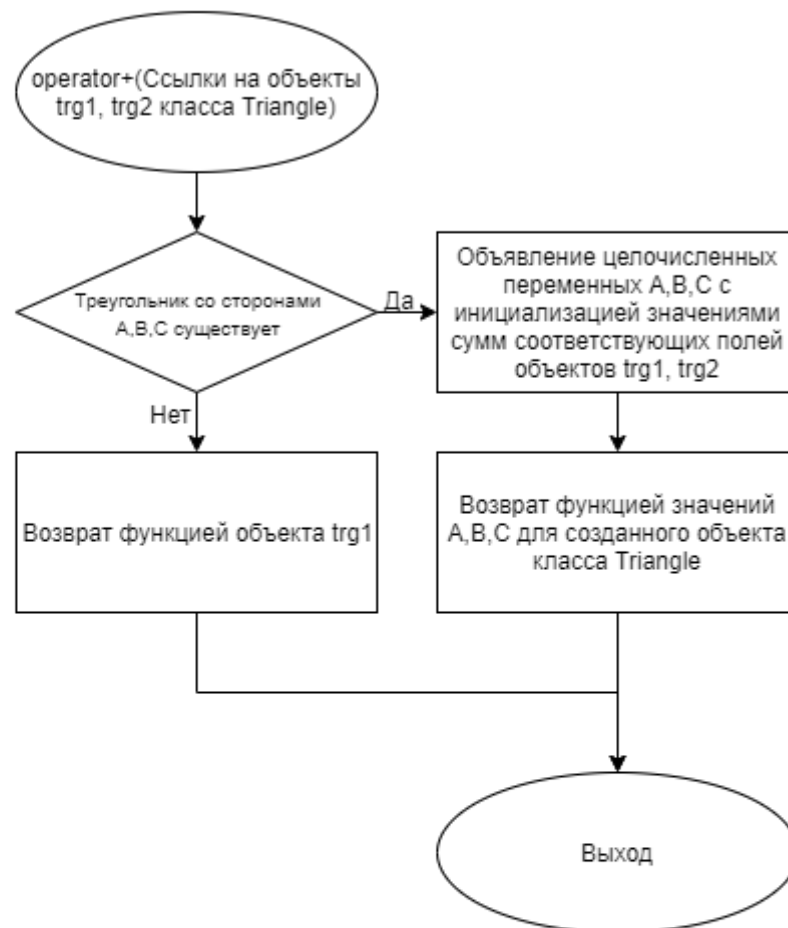


Рис. 6. Блок-схема алгоритма.

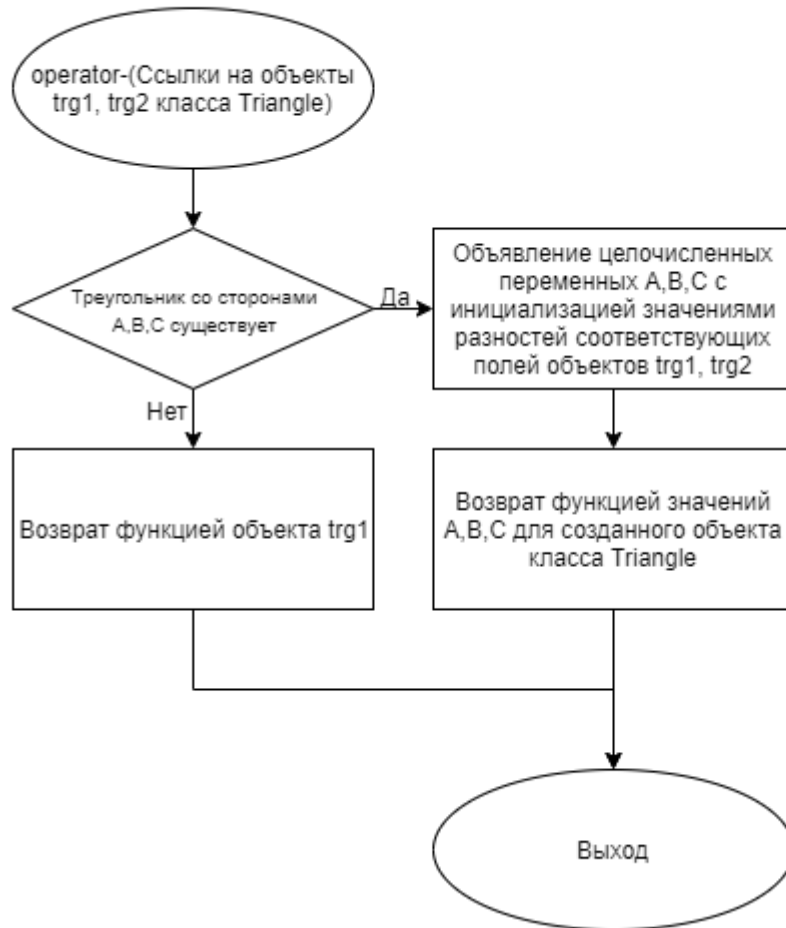


Рис. 7. Блок-схема алгоритма.

## Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

### Файл main.cpp

```
#include "Triangle.h"
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    int n;
    cin>>n;
    vector <Triangle> vec;
    for (int i=0;i<n;i++){
        int a,b,c;
        cin>>a>>b>>c;
        Triangle trg(a,b,c);
        vec.push_back(trg);
    }
    int num1,num2;
    char oper;
    while (cin>>num1>>oper>>num2){
        if (oper=='+') vec[num1-1]=vec[num1-1]+vec[num2-1];
        else vec[num1-1]=vec[num1-1]-vec[num2-1];
    }
    cout<<"a = "<<vec[num1-1].Side1()<<" ; b = "<<vec[num1-1].Side2()<<" ; c
= "<<vec[num1-1].Side3()<<".";
    return 0;
}
```

### Файл Triangle.cpp

```
#include "Triangle.h"
#include <iostream>
#include <cmath>

Triangle :: Triangle(int A, int B, int C){
    a = A;
    b = B;
    c = C;
}

int Triangle :: Perimeter(){
```

```

        int P;
        P = a + b + c;
        return P;
    }

    double Triangle::Square(){
        double S, p;
        p = (a + b + c) / 2.0;
        S = sqrt( p * (p - a) * (p - b) * (p - c) );
        return S;
    }

    int Triangle::Side1() const{
        return a;
    }
    int Triangle::Side2() const{
        return b;
    }
    int Triangle::Side3() const{
        return c;
    }

    Triangle operator+(const Triangle &trg1, const Triangle &trg2){
        int A=trg1.Side1()+trg2.Side1();
        int B=trg1.Side2()+trg2.Side2();
        int C=trg1.Side3()+trg2.Side3();

        if (A+B>C && A+C>B && B+C>A)
            return Triangle(A,B,C);
        else return trg1;
    }

    Triangle operator-(const Triangle &trg1, const Triangle &trg2){
        int A=trg1.Side1()-trg2.Side1();
        int B=trg1.Side2()-trg2.Side2();
        int C=trg1.Side3()-trg2.Side3();

        if (A+B>C && A+C>B && B+C>A && A>0 && B>0 && C>0)
            return Triangle(A,B,C);
        else return trg1;
    }
}

```

## Файл Triangle.h

```

#ifndef _TRIANG_H
#define _TRIANG_H

class Triangle{
private:
    int a;
    int b;
    int c;

public:
    Triangle(int A, int B, int C);
    int Perimeter();

```

```
        double Square();
        int Side1() const;
        int Side2() const;
        int Side3() const;

};

Triangle operator+(const Triangle &trg1, const Triangle &trg2);
Triangle operator-(const Triangle &trg1, const Triangle &trg2);

#endif
```

## Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
3 1 1 1 2 2 2 3 3 3 1+3 1-2	$a = 2; b = 2; c = 2.$	$a = 2; b = 2; c = 2.$
4 5 3 1 2 4 5 3 7 6 1 2 3 2+3 1-2 1-4	$a = 5; b = 3; c = 1.$	$a = 5; b = 3; c = 1.$
4 5 3 1 2 4 5 3 7 6 1 2 3 1-4	$a = 5; b = 3; c = 1.$	$a = 5; b = 3; c = 1.$



## **ЗАКЛЮЧЕНИЕ**

## **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ)**

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.
3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: [https://mirea.aco-avrrora.ru/student/files/methodichescoe\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avrrora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf) (дата обращения 05.05.2021).
4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: [https://mirea.aco-avrrora.ru/student/files/Prilozheniye\\_k\\_methodichke.pdf](https://mirea.aco-avrrora.ru/student/files/Prilozheniye_k_methodichke.pdf) (дата обращения 05.05.2021).
5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».
6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).