



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

Дамарад Д.В.

Руководитель практики

Ассистент

Асадова Ю.С.

Работа представлена

«__»_____ 2022 г.

(подпись студента)

Оценка

(подпись руководителя)

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	
Постановка задачи.....	
Метод решения.....	
Описание алгоритма.....	
Блок-схема алгоритма.....	
Код программы.....	
Тестирование.....	
ЗАКЛЮЧЕНИЕ.....	
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	

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 не дружественных 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 для ввода и вывода на экран.
- Объекты класса Triangle в кол-ве, которое задает пользователь.
- Функции-операторы "+" и "-" для объектов класса Triangle.

Класс Triangle:

- Свойства/поля:
 - Поля, определяющие стороны треугольника:
 - Наименование - a,b,c;
 - Тип - беззнаковый безцелочисленный;
 - Модификатор доступа - private.
- Методы:
 - Метод Triangle:
 - Функционал - параметризованный 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: Целочисленное значение - 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 агт класса vector для хранения объектов класса Triangle	4	
4		Объявление целочисленной переменной с инициализацией i=0	5	
5	i меньше n	Объявление целочисленных беззнаковых переменных a,b,c	6	
			10	
6		Считывание с клавиатуры значений переменных a,b,c	7	

7		Создание объекта <code>trg</code> класса <code>Triangle</code> путем выхода параметризованного конструктора с аргументами <code>a,b,c</code>	8	
8		Вызов метода <code>push_back</code> объекта <code>arr</code> с параметром <code>trg</code>	9	
9		Увеличение <code>i</code> на 1	5	
10		Объявление целочисленных переменных <code>n1,n2</code> и символьной переменной <code>oper</code>	11	
11	Значения <code>n1,oper,n2</code> считаны с клавиатуры	Считывание с клавиатуры значений переменных <code>n1,oper,n2</code>	12	
			13	
12	Значение <code>oper</code> равно '+'	Присвоение значения <code>n1+n2</code> переменной <code>n1</code>	11	
		Присвоение значения <code>n1-n2</code> переменной <code>n1</code>	11	
13		Вывод на экран " <code>a = </code> ", " <code>;</code> ", " <code>b = </code> ", " <code>;</code> ", " <code>c = </code> ", соответствующие значения полей объекта <code>trg</code>	Ø	

Конструктор класса: `Triangle`

Модификатор доступа: `public`

Функционал: Параметризованный 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 <code>A,B,C</code> соответствующим полям <code>a,b,c</code>	Ø	

		объекта класса Triangle		
--	--	-------------------------	--	--

Класс объекта: Triangle

Модификатор доступа: public

Метод: Side1

Функционал: Константный метод, возвращающий длину первой стороны

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 значения поля a объекта класса Triangle	Ø	

Класс объекта: Triangle

Модификатор доступа: public

Метод: Side2

Функционал: Константный метод, возвращающий длину второй стороны

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 значения поля b объекта класса Triangle	Ø	

Класс объекта: Triangle

Модификатор доступа: public

Метод: Side3

Функционал: Константный метод, возвращающий длину третьей стороны

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 значения поля c объекта класса Triangle	Ø	

Функция: operator+

Функционал: Суммирование длин сторон треугольника

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

Функция: operator-

Функционал: Разность длин сторон треугольника

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		Объявление целочисленных переменных А,В,С с инициализацией значениями разностей соответствующих полей объектов trg1, trg2	2	
2	Треугольник со	Возврат функцией	∅	

	сторонами A,B,C существует и длины его сторон больше 0	значений A,B,C для созданного объекта класса Triangle		
		Возврат функцией объекта trg1	∅	

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

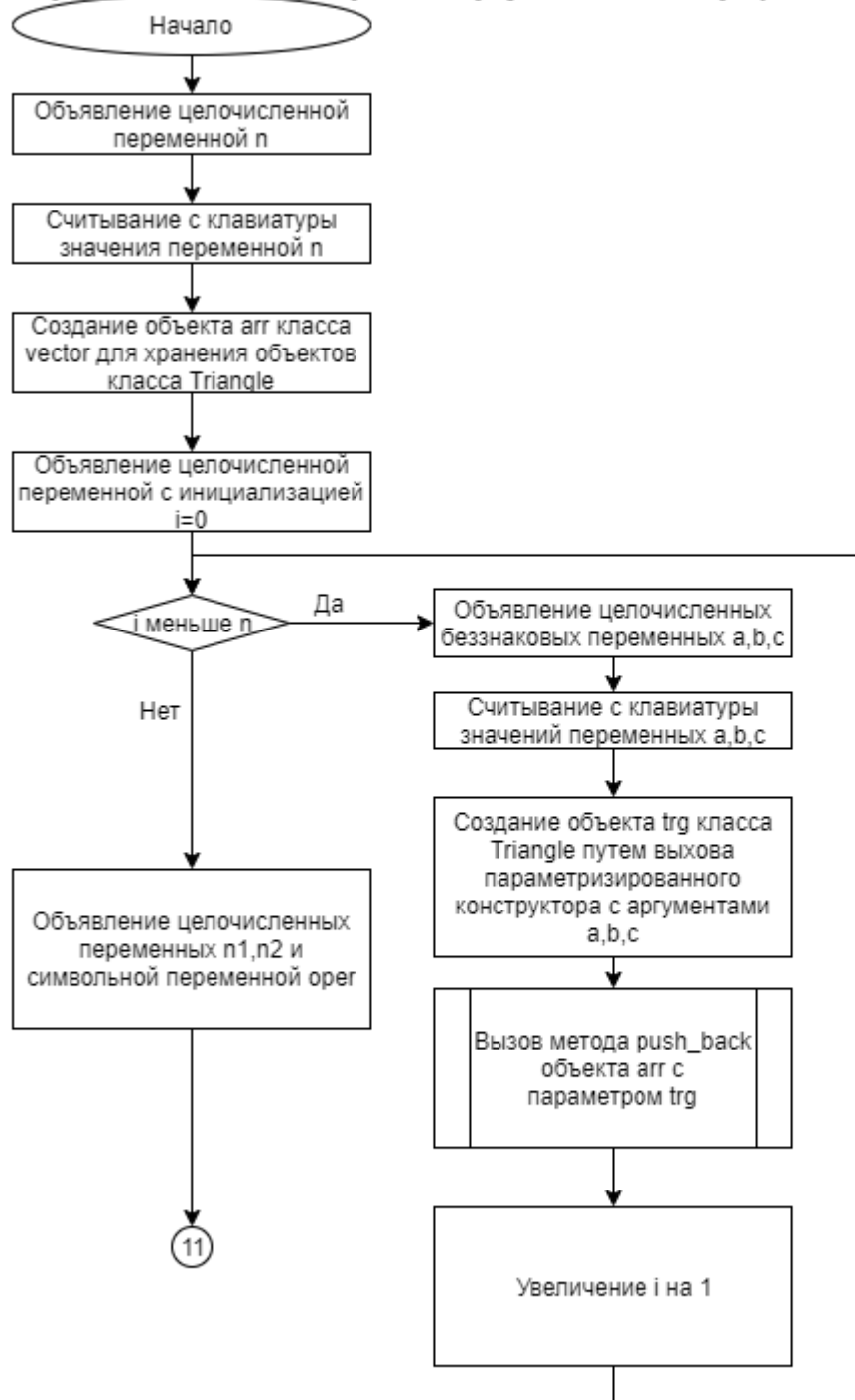


Рис. 1. Блок-схема алгоритма.

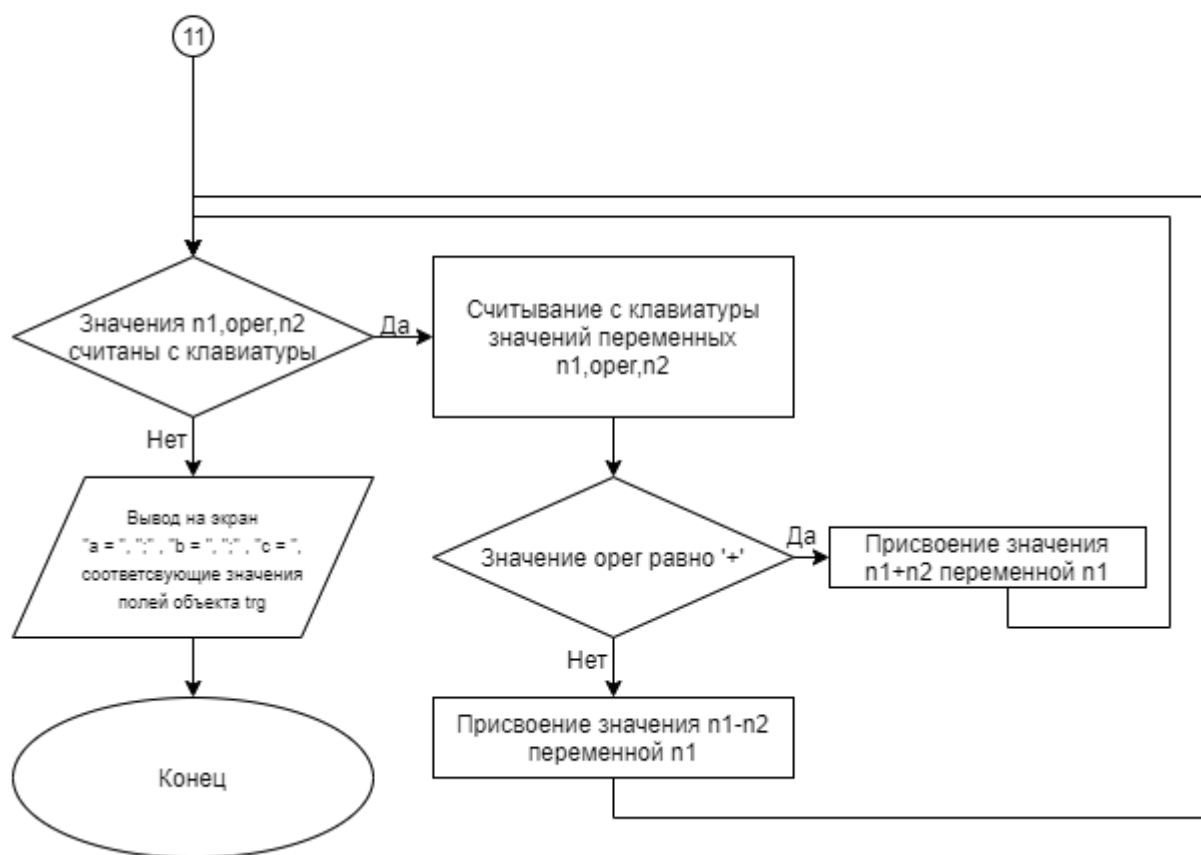


Рис. 2. Блок-схема алгоритма.

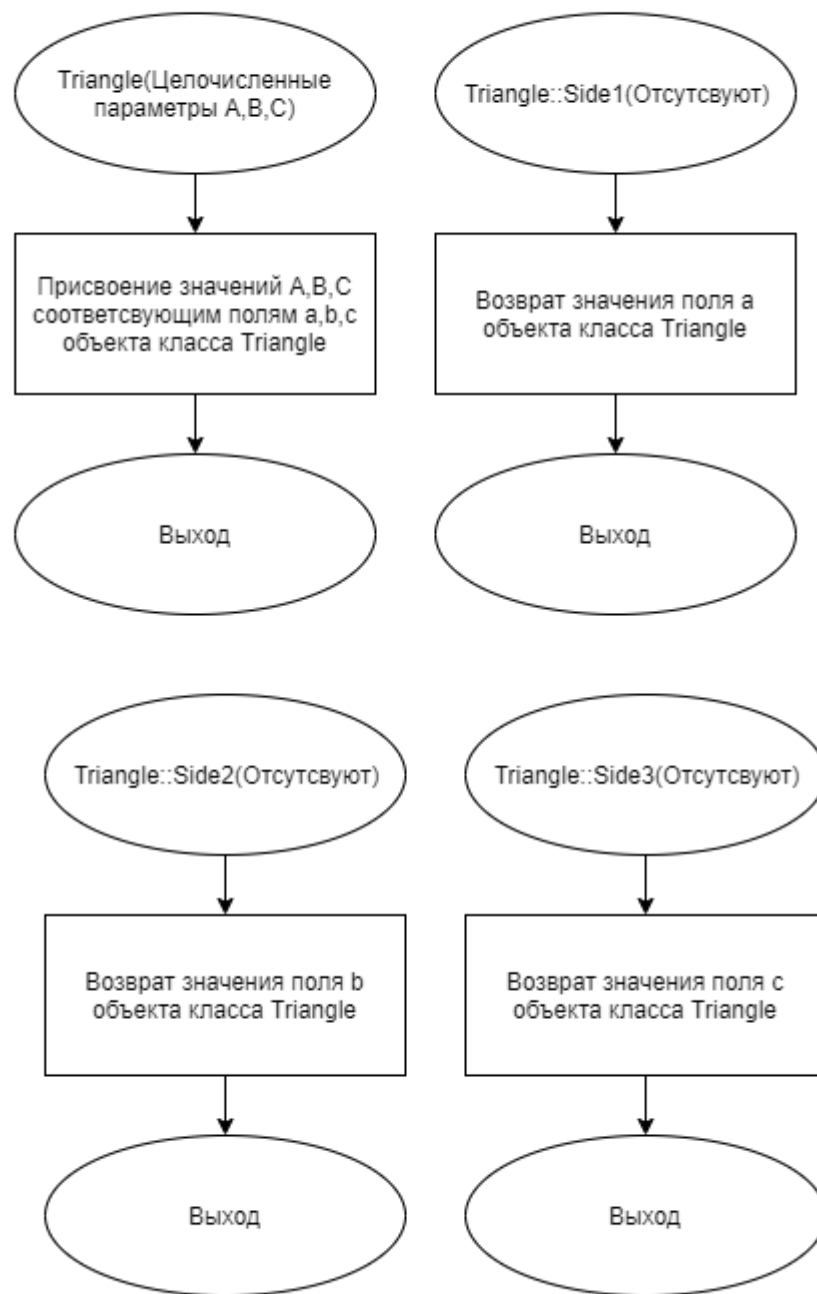


Рис. 3. Блок-схема алгоритма.

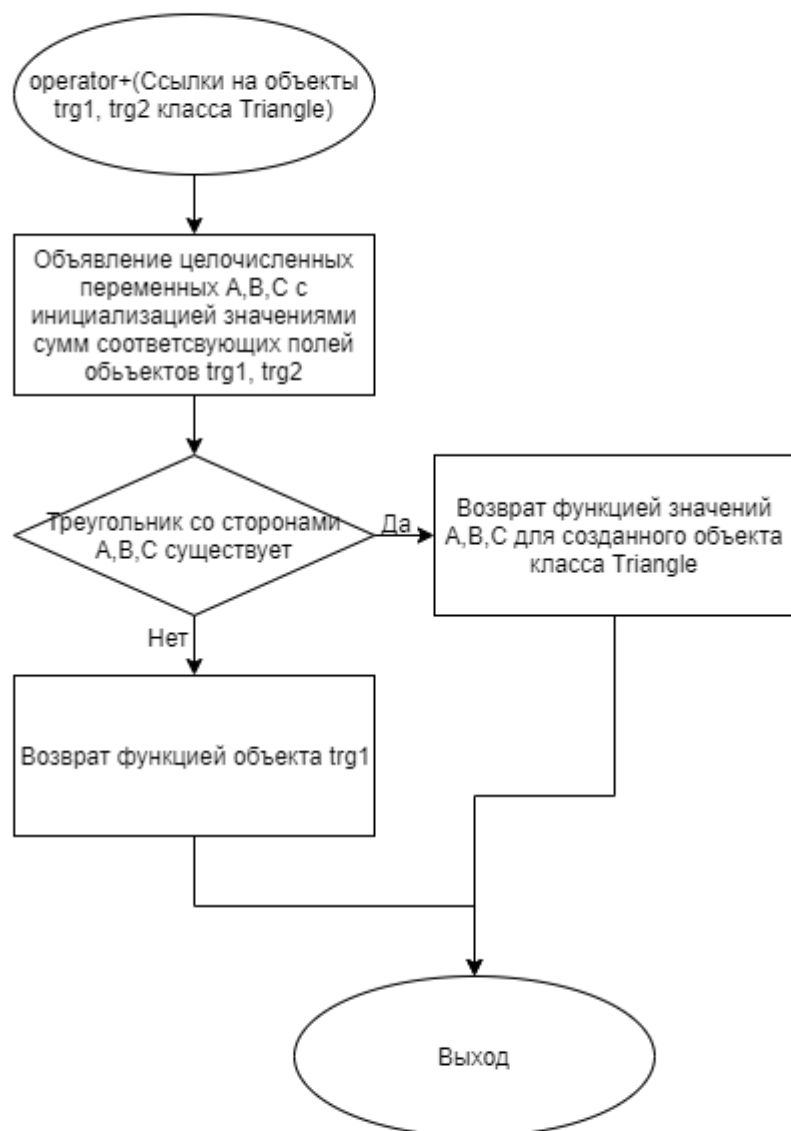


Рис. 4. Блок-схема алгоритма.

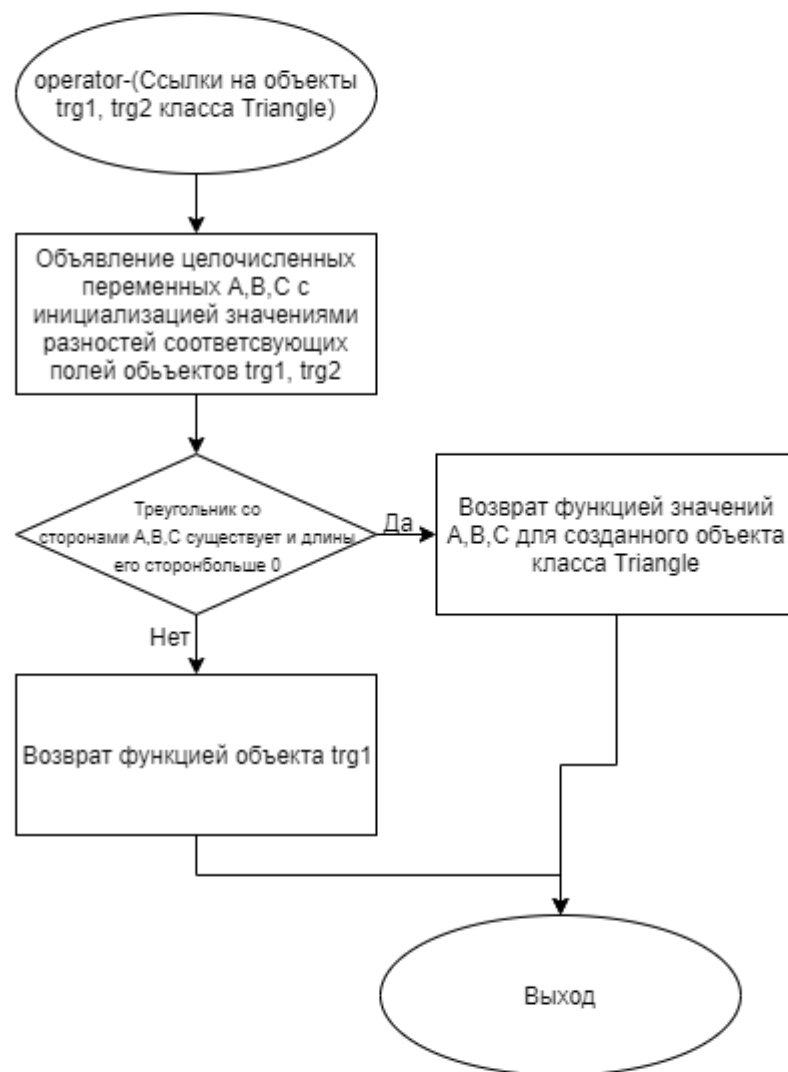


Рис. 5. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл main.cpp

```
#include <iostream>
#include <vector>
#include "triangle.h"
using namespace std;
int main()
{
    int n;
    cin>>n;
    vector<Triangle> arr;
    for (int i=0;i<n;i++){
        unsigned int a,b,c;
        cin>>a>>b>>c;
        Triangle trg(a,b,c);
        arr.push_back(trg);
    }
    int n1,n2;
    char oper;
    while (cin>>n1>>oper>>n2){
        if (oper=='+'){
            arr[n1-1]=arr[n1-1]+arr[n2-1];
        }
        else{
            arr[n1-1]=arr[n1-1]-arr[n2-1];
        }
    }
    cout<<"a = "<<arr[n1-1].Side1()<<" b = "<<arr[n1-1].Side2()<<" c = "<<arr[n1-1].Side3()<<".";
    return 0;
}
```

Файл triangle.cpp

```
#include <math.h>
#include <iostream>
#include "triangle.h"
Triangle::Triangle(int A,int B,int C){
    a=A;
    b=B;
    c=C;
}
int Triangle::P(){
    return a+b+c;
}
```

```

double Triangle::S(){
    double p=(a+b+c)/2.0;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}
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 TRIANGLE_H
#define TRIANGLE_H
class Triangle{
private:
    int a,b,c;
public:
    Triangle(int A,int B,int C);
    int P();
    double S();
    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
2 3 4 5 4 5 6 1 - 2	$a = 3; b = 4; c = 5.$	$a = 3; b = 4; c = 5.$
2 3 4 5 4 5 6 1 + 2	$a = 7; b = 9; c = 11.$	$a = 7; b = 9; c = 11.$

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 (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 (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).