

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	8
3.1 Алгоритм метода operator+ класса Triangle.....	8
3.2 Алгоритм метода operator- класса Triangle.....	8
3.3 Алгоритм метода Get_a класса Triangle.....	9
3.4 Алгоритм метода Get_b класса Triangle.....	9
3.5 Алгоритм метода Get_c класса Triangle.....	10
3.6 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	16
5.1 Файл main.cpp.....	16
5.2 Файл Triangle.cpp.....	17
5.3 Файл Triangle.h.....	18
6 ТЕСТИРОВАНИЕ.....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	20

1 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.

1.1 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»

1.2 Описание выходных данных

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

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект второй треугольник класса `ob` предназначен для операции;
- функция `vector` для динамический массив;
- `While` - оператор цикла с условием;
- `if` - условный оператор.

Класс `Triangle`:

- свойства/поля:
 - поле Сторона `a`:
 - наименование — `a`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле Сторона `b`:
 - наименование — `b`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле Сторона `c`:
 - наименование — `c`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - метод `Get_a` — Возвращение значение `a`;
 - метод `Get_b` — Возвращение значение `b`;
 - метод `Get_c` — Возвращение значение `c`;
 - метод `operator+` — Переопределение операции сложения;
 - метод `operator-` — Переопределение операции вычитания.

3 ОПИСАНИЕ АЛГОРИТМОВ

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

3.1 Алгоритм метода `operator+` класса `Triangle`

Функционал: Переопределение операции сложения.

Параметры: `Triangle ob` - второй треугольник.

Возвращаемое значение: `Triangle` - возвращает объект класса `Triangle`.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода `operator+` класса `Triangle`

№	Предикат	Действия	№ перехода
1		инициализация <code>a1, b1, c1</code> = стороны объекта <code>ob</code>	2
2	Треугольник со сторонами - суммами треугольников существует	<code>a+=a1; b+=b1; c+=c1.</code>	3
			3
3		вернуть <code>this*</code>	Ø

3.2 Алгоритм метода `operator-` класса `Triangle`

Функционал: Переопределение операции вычитания.

Параметры: `Triangle ob` - второй треугольник.

Возвращаемое значение: `Triangle` - возвращает объект класса `Triangle`.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода operator- класса Triangle

№	Предикат	Действия	№ перехода
1		инициализация a1, b1, c1 = стороны объекта ob	2
2	Разности сторон треугольников больше нуля		3
			4
3	Треугольник со сторонами - разностями треугольников существует	a-=a1; b-=b1; c-=c1.	4
			4
4		вернуть this*	Ø

3.3 Алгоритм метода Get_a класса Triangle

Функционал: Возвращение значение a.

Параметры: нет.

Возвращаемое значение: int - сторона.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода Get_a класса Triangle

№	Предикат	Действия	№ перехода
1		Возвращение значение a	Ø

3.4 Алгоритм метода Get_b класса Triangle

Функционал: Возвращение значение b.

Параметры: нет.

Возвращаемое значение: int - сторона.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *Get_b* класса *Triangle*

№	Предикат	Действия	№ перехода
1		Возвращение значение b	Ø

3.5 Алгоритм метода *Get_c* класса *Triangle*

Функционал: Возвращение значение c.

Параметры: нет.

Возвращаемое значение: int - сторона.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *Get_c* класса *Triangle*

№	Предикат	Действия	№ перехода
1		Возвращение значение c	Ø

3.6 Алгоритм функции *main*

Функционал: Основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: int - индикатор корректности завершения алгоритма.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Ввести количество треугольников n с клавиатуры	2
2		В массив Triangles записать созданные по введенным с клавиатуры сторонам треугольники	3
3		Выполнить введенные с клавиатуры операции сложения и/или	4

№	Предикат	Действия	№ перехода
		вычитания	
4		вывести на экран результат действия последней операции	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

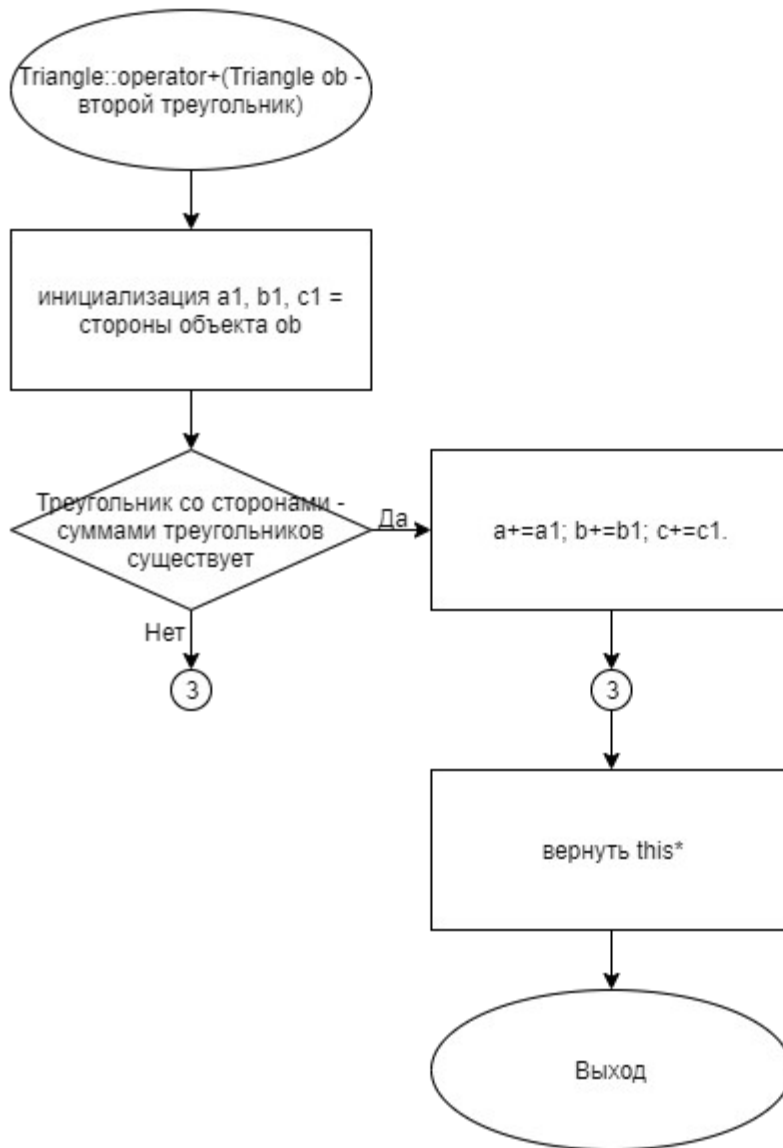


Рисунок 1 – Блок-схема алгоритма

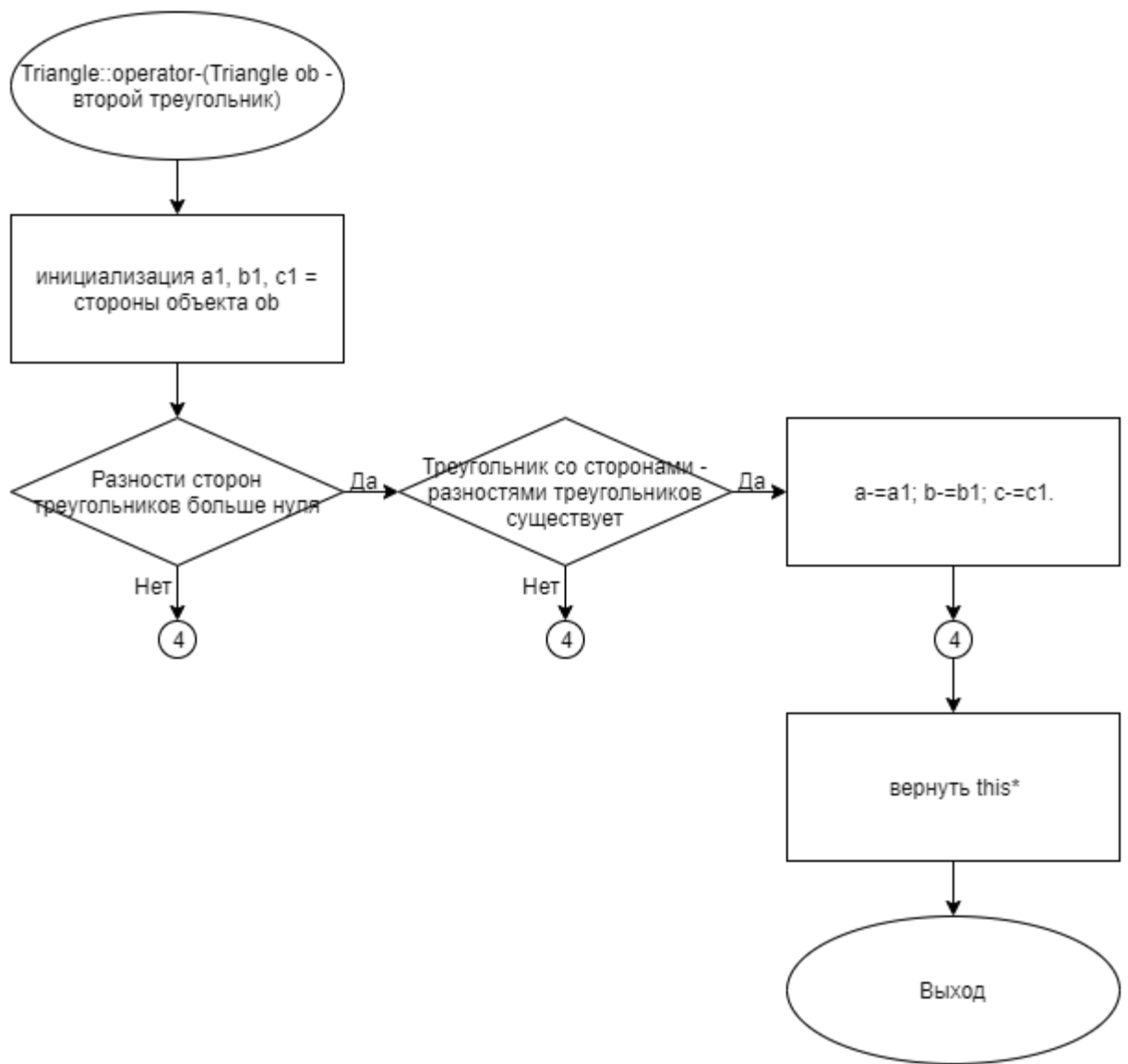


Рисунок 2 – Блок-схема алгоритма

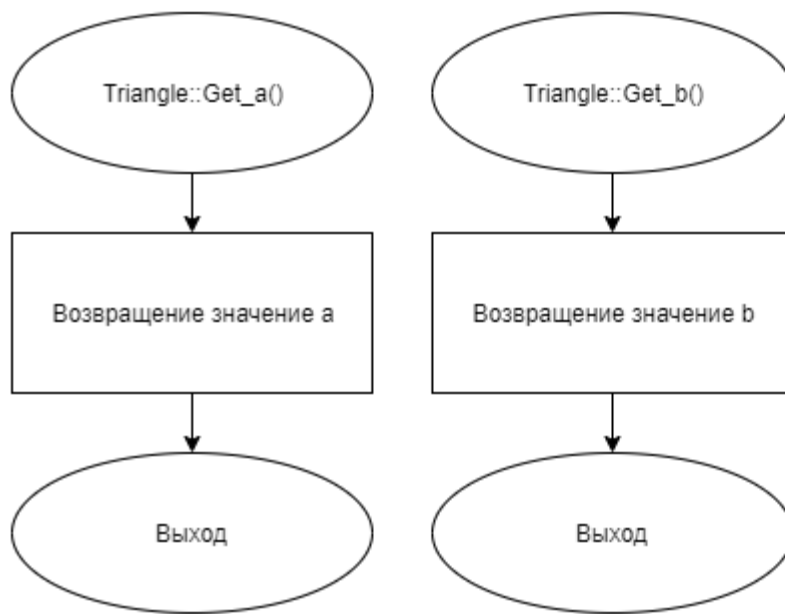


Рисунок 3 – Блок-схема алгоритма



Рисунок 4 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <vector>
#include "Triangle.h"
using namespace std;
int main()
{
    int n,x,y,a,b,c;
    char com;
    vector <int> Side;
    vector <Triangle> Triangles;
    cin>>n;
    for (int i =0;i<n;i++){
        cin>>a>>b>>c;
        Side.push_back(a);
        Side.push_back(b);
        Side.push_back(c);

        Triangles.push_back(Triangle(Side[0+(3*i)],Side[1+(3*i)],Side[2+(3*i)]));
    }

    while (cin>>x){
        cin>>com>>y;
        if (com=='+') Triangles[x-1]=Triangles[x-1]+Triangles[y-1];
        else if (com=='-') Triangles[x-1]=Triangles[x-1]-Triangles[y-1];
    }

    a = Triangles[x-1].Get_a();
    b = Triangles[x-1].Get_b();
    c = Triangles[x-1].Get_c();
    cout<<"a = "<<a<<" b = "<<b<<" c = "<<c<<".";
    return(0);
}
```

5.2 Файл Triangle.cpp

Листинг 2 – Triangle.cpp

```
#include "Triangle.h"
#include <iostream>
#include <math.h>
Triangle::Triangle(int a1, int b1, int c1){
    a=a1;
    b=b1;
    c=c1;
}
double Triangle::Per(){
    double P = a+b+c;
    return P;
}
double Triangle::Square(double P){
    double P1=P/2;
    double S = sqrt(P1*(P1-a)*(P1-b)*(P1-c));
    return S;
}
int Triangle::Get_a(){
    return a;
}
int Triangle::Get_b(){
    return b;
}
int Triangle::Get_c(){
    return c;
}
Triangle Triangle::operator+(Triangle ob){
    int a1,b1,c1;
    a1 = ob.Get_a();
    b1 = ob.Get_b();
    c1 =ob.Get_c();
    if(a1+a+b1+b>c1+c && a1+a+c1+c >b1+b && b1+b+c1+c>a1+a){
        a+=a1;
        b+=b1;
        c+=c1;
    }
    return *this;
}

Triangle Triangle::operator-(Triangle ob){
    int a1,b1,c1;
    a1 = ob.Get_a();
    b1 = ob.Get_b();
    c1 =ob.Get_c();
    if (a-a1>0&&b-b1>0&&c-c1>0){
        if(a-a1+b-b1>c-c1 && a-a1+c-c1 >b-b1 && b-b1+c-c1>a-a1){
            a-=a1;
            b-=b1;
            c-=c1;
        }
    }
}
```

```
    }  
    return *this;  
}  
Triangle::~Triangle(){}  
}
```

5.3 Файл Triangle.h

Листинг 3 – Triangle.h

```
#ifndef __TRIANGLE__H  
#define __TRIANGLE__H  
class Triangle  
{  
    private:  
        int a;  
        int b;  
        int c;  
    public:  
        Triangle(int a1, int b1, int c1);  
        int Get_a();  
        int Get_b();  
        int Get_c();  
        double Per( );  
        Triangle operator+(Triangle ob);  
        Triangle operator-(Triangle ob);  
        double Square(double P) ;  
        ~Triangle();  
};  
#endif
```


6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 1 2 3 4 5 6 1 + 2	$a = 5; b = 7; c = 9.$	$a = 5; b = 7; c = 9.$

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации.
2. Методическое пособие студента для выполнения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avvora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
3. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avvora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
4. Шилдт Г. С++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2019. — 624 с.
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).