



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-27-21

Шевелёв И.А.

Руководитель практики

Ассистент

Морозов В.А.

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.....	14
Код программы.....	19
Тестирование.....	21
ЗАКЛЮЧЕНИЕ.....	22
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	23

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

Основная программа:

Целочисленный тип данных

Объекты ввода/вывода потока данных (cin/cout библиотеки <iostream>)

Вектор библиотеки <vector>

Цикл while, for

Символьный тип данных

Условный оператор if

Объект temp класса Triangles

Класс Triangles:

Модификатор доступа public

Целочисленный тип данных

Конструктор - Для возможности объявление объекта

Конструктор копирования - копирует объект

Параметризированный конструктор - передачи значений в свойства объекта

Метод operator- - перегружает оператор минус

Метод operator+ - перегружает оператор плюс

Номер класса	Имя класса	Модификатор доступа	Описание	Номер	Коментарий
--------------	------------	---------------------	----------	-------	------------

1	Triangles	public	Треугольник		
---	-----------	--------	-------------	--	--

Описание алгоритма

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

Конструктор класса: Triangles

Модификатор доступа: public

Функционал: Конструктор копирования

Параметры: const Triangles &other

Алгоритм конструктора представлен в таблице 2.

Таблица 2. Алгоритм конструктора класса Triangles

№	Предикат	Действия	№ перехода	Комментарий
1		Присвоение свойству a текущего объекта значение a объекта other	2	
2		Присвоение свойству b текущего объекта значение b объекта other	3	
3		Присвоение свойству c текущего объекта значение c объекта other	Ø	

Конструктор класса: Triangles

Модификатор доступа: public

Функционал: Конструктор

Параметры: int a, int b, int c

Алгоритм конструктора представлен в таблице 3.

Таблица 3. Алгоритм конструктора класса Triangles

№	Предикат	Действия	№ перехода	Комментарий
1		Присвоение свойству a значение параметра a	2	
2		Присвоение свойству b значение параметра b	3	
3		Присвоение свойству c значение параметра c	∅	

Функция: operator +

Функционал: Перегрузка оператора +

Параметры: Triangles other1, Triangles other2

Возвращаемое значение: Triangles, temp

Алгоритм функции представлен в таблице 4.

Таблица 4. Алгоритм функции operator +

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление временного объекта temp	2	
2		Свойству a объекта temp присваиваем значение other1.a + other2.a	3	
3		Свойству b объекта temp присваиваем значение other1.b + other2.b	4	
4		Свойству c объекта temp присваиваем значение other1.c + other2.c	5	
5	temp.a + temp.b >	Вернуть temp	∅	

	temp.c && temp.a + temp.c > temp.b && temp.b + temp.c > temp.a			
			6	
6		Свойству a объекта temp присваиваем значение other1.a	7	
7		Свойству b объекта temp присваиваем значение other1.b	8	
8		Свойству c объекта temp присваиваем значение other1.c	9	
9		Вернуть temp	Ø	

Функция: operator -

Функционал: Перегрузка оператора +

Параметры: Triangles other1, Triangles other2

Возвращаемое значение: Triangles, temp

Алгоритм функции представлен в таблице 5.

Таблица 5. Алгоритм функции operator -

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление временного объекта temp	2	
2		Свойству a объекта temp присваиваем значение other1.a - other2.a	3	
3		Свойству b объекта temp присваиваем значение other1.b - other2.b	4	

4		Свойству с объекта temp присваиваем значение other1.c - other2.c	5	
5	temp.a + temp.b > temp.c && temp.a + temp.c > temp.b && temp.b + temp.c > temp.a	Вернуть temp	Ø	
			6	
6		Свойству a объекта temp присваиваем значение other1.a	7	
7		Свойству b объекта temp присваиваем значение other1.b	8	
8		Свойству c объекта temp присваиваем значение other1.c	9	
9		Вернуть temp	Ø	

Функция: main

Функционал: Главная функция программы

Параметры: нет

Возвращаемое значение: int, Код возврата

Алгоритм функции представлен в таблице 6.

Таблица 6. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленных переменных n, a, b, c, num1, num2	2	

2		Объявление переменной символьного типа данных oper	3	
3		Ввод значения n	4	
4		Объявление объекта mas класса vector	5	
5	i < n	Инкремент i	6	
			9	
6		Ввод значений a, b, c	7	
7		Создание объекта temp с параметрами a, b, c класса Triangles	8	
8		Добавление в массив mas объект temp через конструктор копирования	5	
9	cin >> num1 >> oper >> num2		10	
			11	
10	oper == '+'	Присвоение элементу массива mas[num1-1] значение mas[num1 - 1] + mas[num2 - 1]	9	
	oper == '-'	Присвоение элементу массива mas[num1-1] значение mas[num1 - 1] - mas[num2 - 1]	9	
11		Вывод "a = " mas[num1 - 1].a "; "	12	
12		Вывод "a = " mas[num1 - 1].b "; "	13	
13		Вывод "a = " mas[num1 - 1].c "; "	Ø	

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

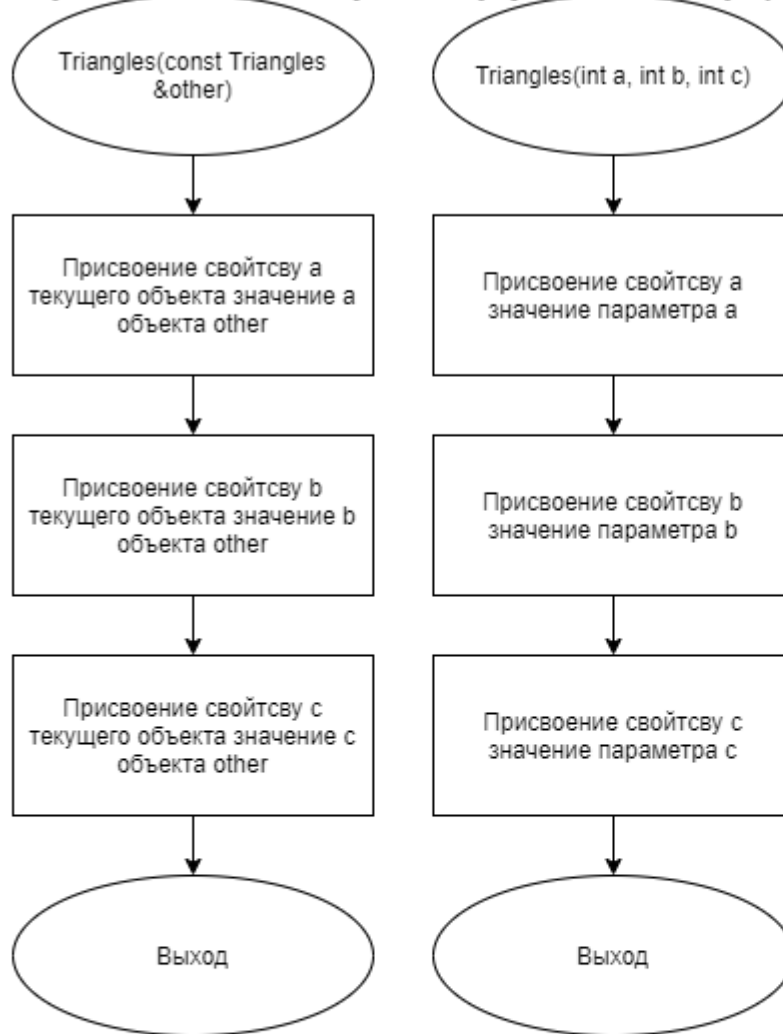


Рис. 1. Блок-схема алгоритма.

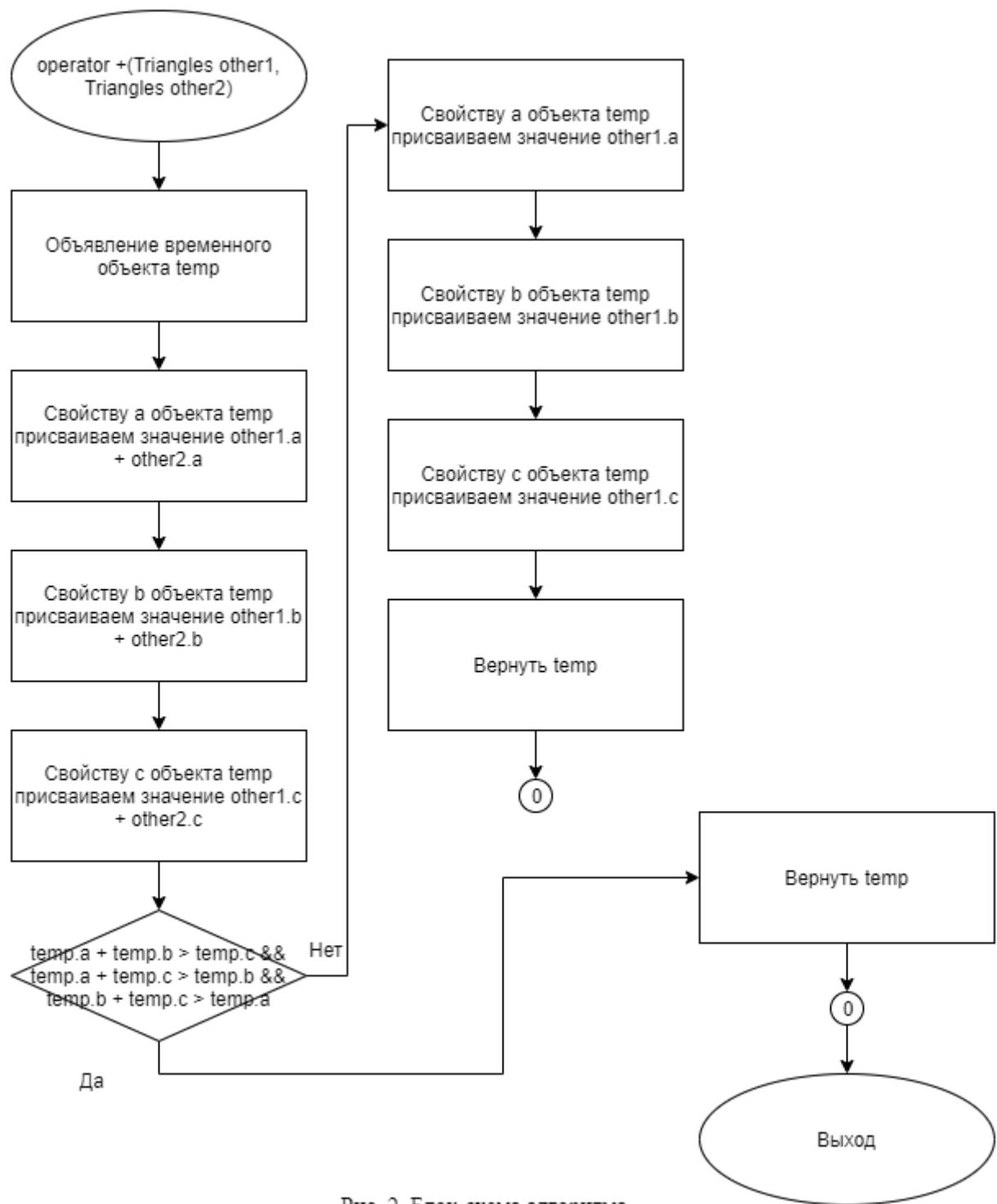


Рис. 2. Блок-схема алгоритма.

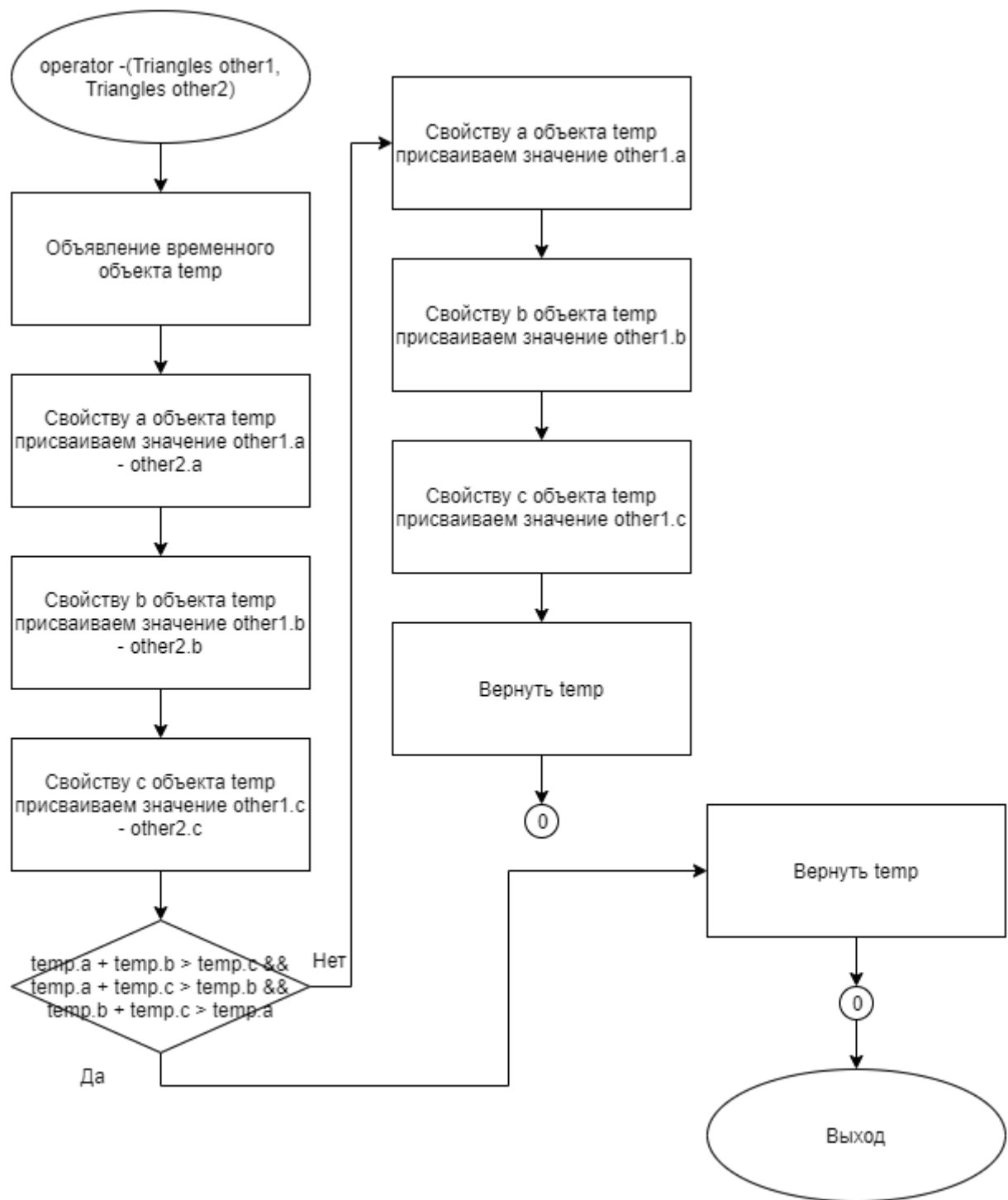


Рис. 3. Блок-схема алгоритма.

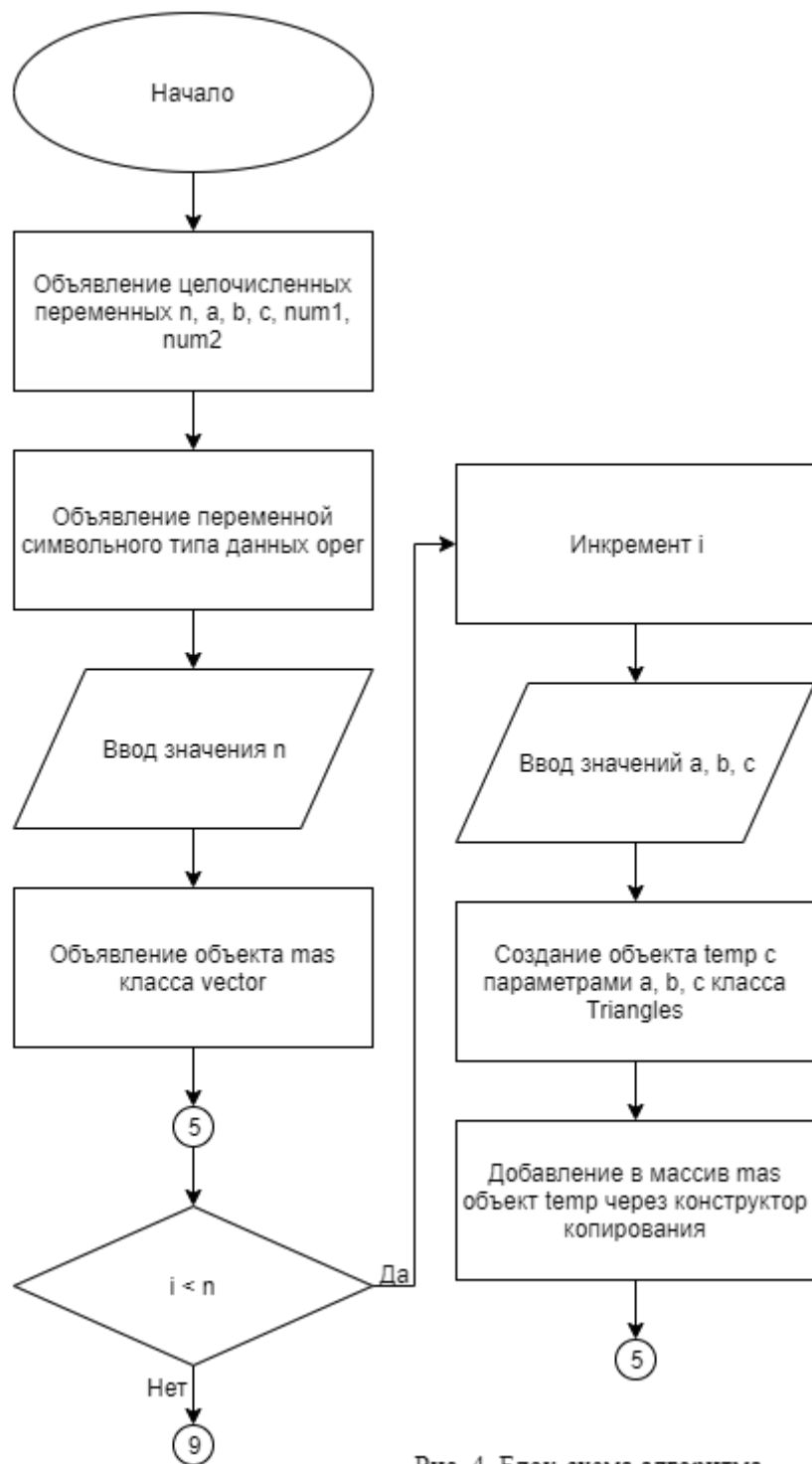


Рис. 4. Блок-схема алгоритма.

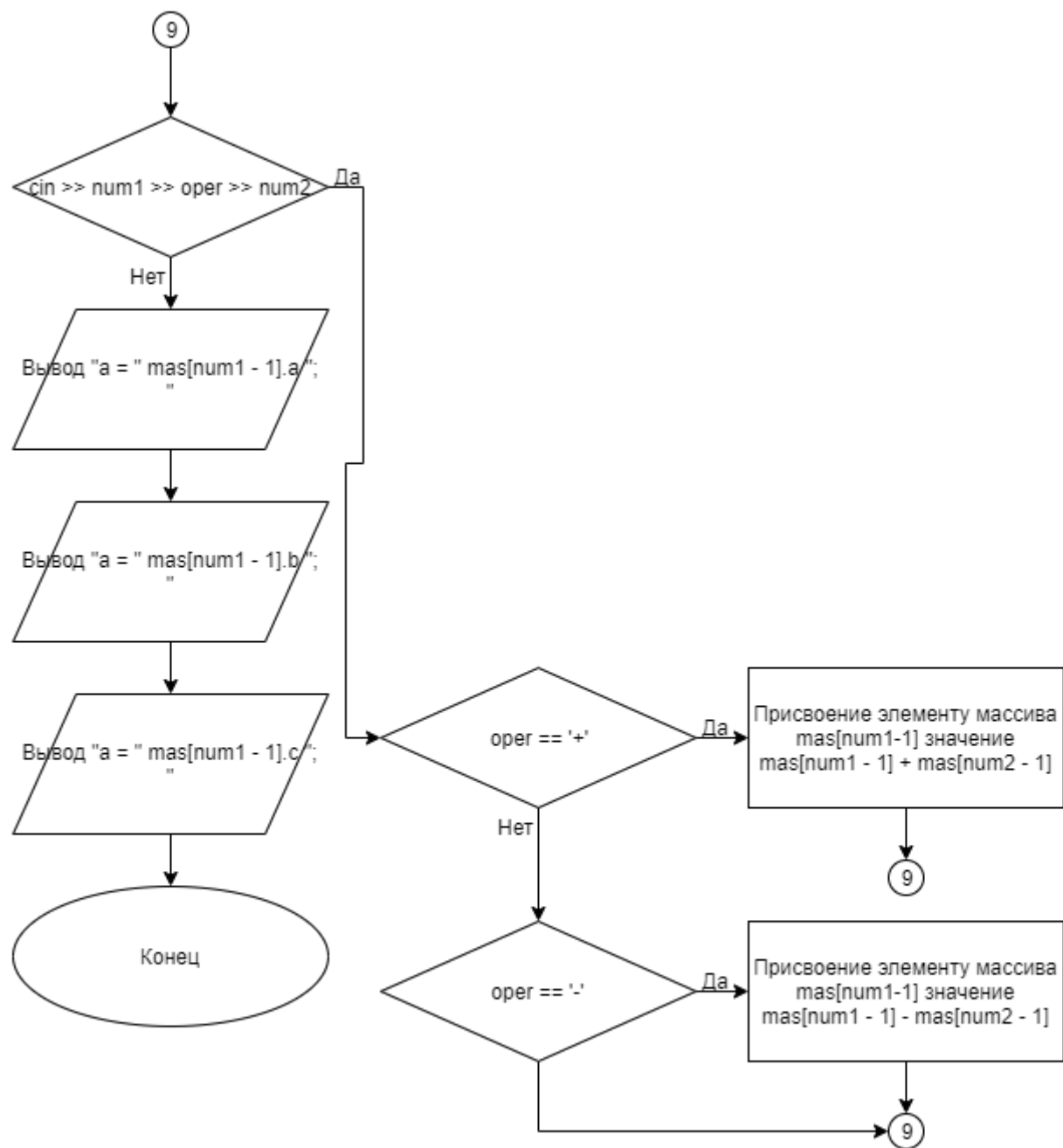


Рис. 5. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "trey.h"

const Triangles operator +(Triangles other1, Triangles other2){
    Triangles temp;
    temp.a = other1.a + other2.a;
    temp.b = other1.b + other2.b;
    temp.c = other1.c + other2.c;
    if(temp.a + temp.b > temp.c && temp.a + temp.c > temp.b && temp.b +
temp.c > temp.a){
        return temp;
    }else{
        temp.a = other1.a;
        temp.b = other1.b;
        temp.c = other1.c;
        return temp;
    }
}

const Triangles operator -(Triangles other1, Triangles other2){
    Triangles temp;
    temp.a = other1.a - other2.a;
    temp.b = other1.b - other2.b;
    temp.c = other1.c - other2.c;
    if(temp.a + temp.b > temp.c && temp.a + temp.c > temp.b && temp.b +
temp.c > temp.a){
        return temp;
    }else{
        temp.a = other1.a;
        temp.b = other1.b;
        temp.c = other1.c;
        return temp;
    }
}

int main()
{
    // program here
    int n, a, b, c;
    int num1, num2;
    char oper;

    cin >> n;
    vector<Triangles> mas;

    for(int i = 0; i < n; i++){
```

```

        cin >> a >> b >> c;
        Triangles temp(a, b, c);
        mas.push_back(Triangles(temp));
    }

    while(cin >> num1 >> oper >> num2){
        if(oper == '+')
            mas[num1-1] = mas[num1 - 1] + mas[num2 - 1];
        else
            mas[num1-1] = mas[num1 - 1] - mas[num2 - 1];
    }
    cout << "a = " << mas[num1-1].a << "; ";
    cout << "b = " << mas[num1-1].b << "; ";
    cout << "c = " << mas[num1-1].c << ".";
    return(0);
}

```

Файл trey.cpp

```

#include "trey.h"
Triangles::Triangles(const Triangles &other){
    this->a = other.a;
    this->b = other.b;
    this->c = other.c;
}

```

Файл trey.h

```

#ifndef trey_h
#define trey_h
#include <iostream>
#include <cmath>
#include <vector>
using namespace std;

class Triangles{
public:
    int a, b, c;
    Triangles() {}
    Triangles(int a,int b,int c) : a(a), b(b), c(c) {};
    Triangles(const Triangles &other);
};
#endif

```

Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 1 2 3 3 1 1 1 + 2	a = 4; b = 3; c = 4.	a = 4; b = 3; c = 4.
3 1 22 3 12 32 43 86 45 86 1 + 2 2 - 1 3 + 1	a = 99; b = 99; c = 132.	a = 99; b = 99; c = 132.
2 1 2 3 3 2 1 2 - 1	a = 3; b = 2; c = 1.	a = 3; b = 2; c = 1.

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ)

1. Васильев А.Н. Объектно-ориентированное программирование на С++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. С++: 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).