Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

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. a1 + a2, b1 + b2, c1 + c2. 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:

- функция main для основная функция программы;
- функция operator+ для оператор сложения;
- функция operator- для оператор вычитания;
- cin - объект стандартного потока ввода с клавиатуры;
- cout - объект стандартного потока вывода на экран;
- if .. else - условный оператор;
- for - оператор со счётчиком;
- while - оператор цикла с предусловием;
- push_back - для вставки нового объекта "треугольник" в конец вектора "треугольники".

Класс triangle:

- свойства/поля:
    - поле значение стороны a треугольника:
        - наименование — a;
        - тип — int;
        - модификатор доступа — private;
    - поле значение стороны b треугольника:
        - наименование — b;
        - тип — int;
        - модификатор доступа — private;
    - поле значение стороны c треугольника:
        - наименование — c;
        - тип — int;
        - модификатор доступа — private;

- функционал:
  - метод triangle — параметризированный конструктор;
  - метод result — выводит значения полей объекта на экран;
  - метод geta — получение значения поля a объекта;
  - метод getb — получение значения поля b объекта;
  - метод getc — получение значения поля c объекта.

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 конструктора класса triangle

Функционал: параметризированный конструктор.

Параметры: int a, int b, int c.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса triangle

№	Предикат	Действия	№ перехода
1		присваивание полю данного объекта a значение параметра a	2
2		присваивание полю данного объекта b значение параметра b	3
3		присваивание полю данного объекта c значение параметра c	∅

3.2 Алгоритм метода result класса triangle

Функционал: выводит значения полей объекта на экран.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода result класса triangle

№	Предикат	Действия	№ перехода
1		вывод на экран "a = ", значение поля a данного объекта, "; b = ", значение поля b данного объекта, "; c = " значение поля c данного объекта и '.'	∅

3.3 Алгоритм функции operator+

Функционал: оператор сложения.

Параметры: triangle& t1, triangle& t2.

Возвращаемое значение: triangle.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции operator+

№	Предикат	Действия	№ перехода
1		инициализация целочисленных переменных a1, a2, b1, b2, c1, c2	2
2		присваивание переменной a1 результат метода geta объекта t1, присваивание переменной b1 результат метода getb объекта t1, присваивание переменной c1 результат метода getc объекта t1	3
3		присваивание переменной a2 результат метода geta объекта t2, присваивание переменной b2 результат метода getb объекта t2, присваивание переменной c2 результат метода getc объекта t2	4
4		инициализация переменной at типа int значением результата суммы переменных a1 и a2	5
5		инициализация переменной bt типа int значением результата суммы переменных b1 и b2	6
6		инициализация переменной ct типа int значением результата суммы переменных c1 и c2	7
7	at + bt > ct && at + ct > bt && bt + ct > at	возврат объекта, созданного с помощью параметризированного конструктора с аргументами at, bt и ct	∅
		возврат указателя на объект t1	∅

3.4 Алгоритм функции operator-

Функционал: оператор вычитания.

Параметры: triangle& t1, triangle& t2.

Возвращаемое значение: triangle.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции operator-

№	Предикат	Действия	№ перехода
1		инициализация целочисленных переменных a1 и a2 значениями результатов методов geta объектов t1 и t2 соответственно	2
2		инициализация переменной at типа int значеним результата разности переменных a1 и a2	3
3	at > 0	инициализация целочисленных переменных b1 и b2 значениями результатов методов geta объектов t1 и t2 соответственно	4
			9
4		инициализация переменной bt типа int значением результата разности переменных b1 и b2	5
5	bt > 0	инициализация целочисленных переменных c1 и c2 значениями результатов методов geta объектов t1 и t2 соответственно	6
			9
6		инициализация переменной ct типа int значением результата разности переменных c1 и c2	7
7	ct > 0		8
			9
8	at + bt > ct && at + ct > bt && bt + ct > at	возврат объекта, созданного с помощью параметризированного конструктора с	∅

№	Предикат	Действия	№ перехода
		аргументами at, bt и ct	
			9
9		возврат указателя на объект t1	∅

3.5 Алгоритм функции main

Функционал: основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		инициализация переменной n типа int	2
2		ввод значения переменной n с клавиатуры	3
3		создание вектора triangles с данными класса triangle	4
4		инициализация переменной i типа int значением 0	5
5	i < n	инициализация переменных a, b и c типа int	6
			8
6		ввод значений переменных a, b и c с клавиатуры	7
7		i++	5
8		инициализация переменных triangle1, triangle2 типа int и переменной action типа char	9
9	cin >> triangle1 >> action >> triangle2		10
		вызов метода result() объекта вектора triangles с индексом triangle1 - 1	11
10	action == '+'	вычисление суммы объекта вектора triangles с	9

№	Предикат	Действия	№ перехода
		индексом triangle1 - 1 и объекта вектора triangles с индексом triangle2 - 1	
		вычисление разности объекта вектора triangles с индексом triangle1 - 1 и объекта вектора triangles с индексом triangle2 - 1	9
11		возврат значения 0	∅

3.6 Алгоритм метода geta класса triangle

Функционал: получение значения поля a объекта.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода geta класса triangle

№	Предикат	Действия	№ перехода
1		возврат значения поля a объекта	∅

3.7 Алгоритм метода getb класса triangle

Функционал: получение значения поля b объекта.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода getb класса triangle

№	Предикат	Действия	№ перехода
1		возврат значения поля b объекта	∅

3.8 Алгоритм метода getc класса triangle

Функционал: получение значения поля с объекта.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода getc класса triangle

№	Предикат	Действия	№ перехода
1		возврат значения поля с объекта	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.

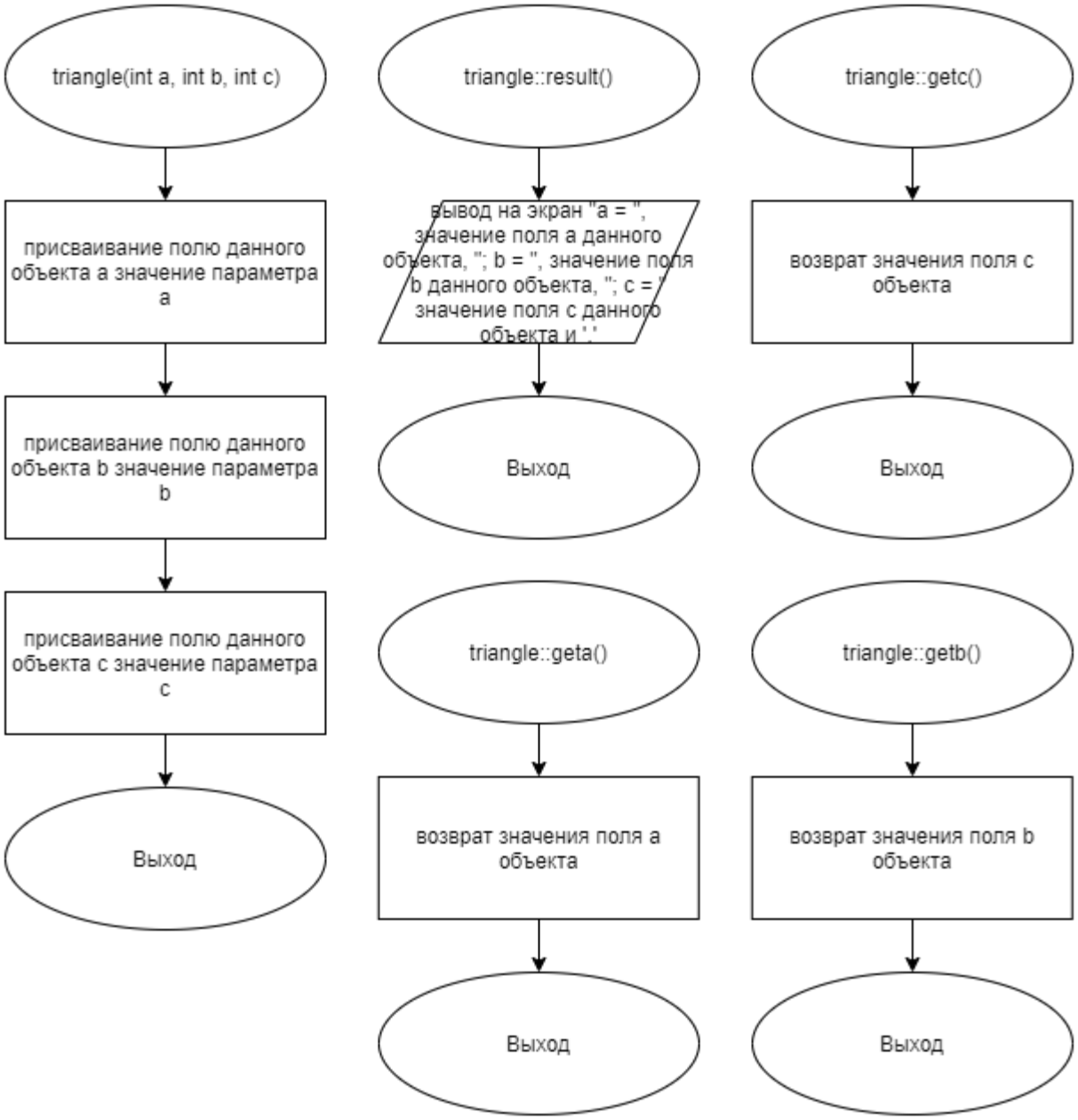

Рисунок 1 – Блок-схема алгоритма

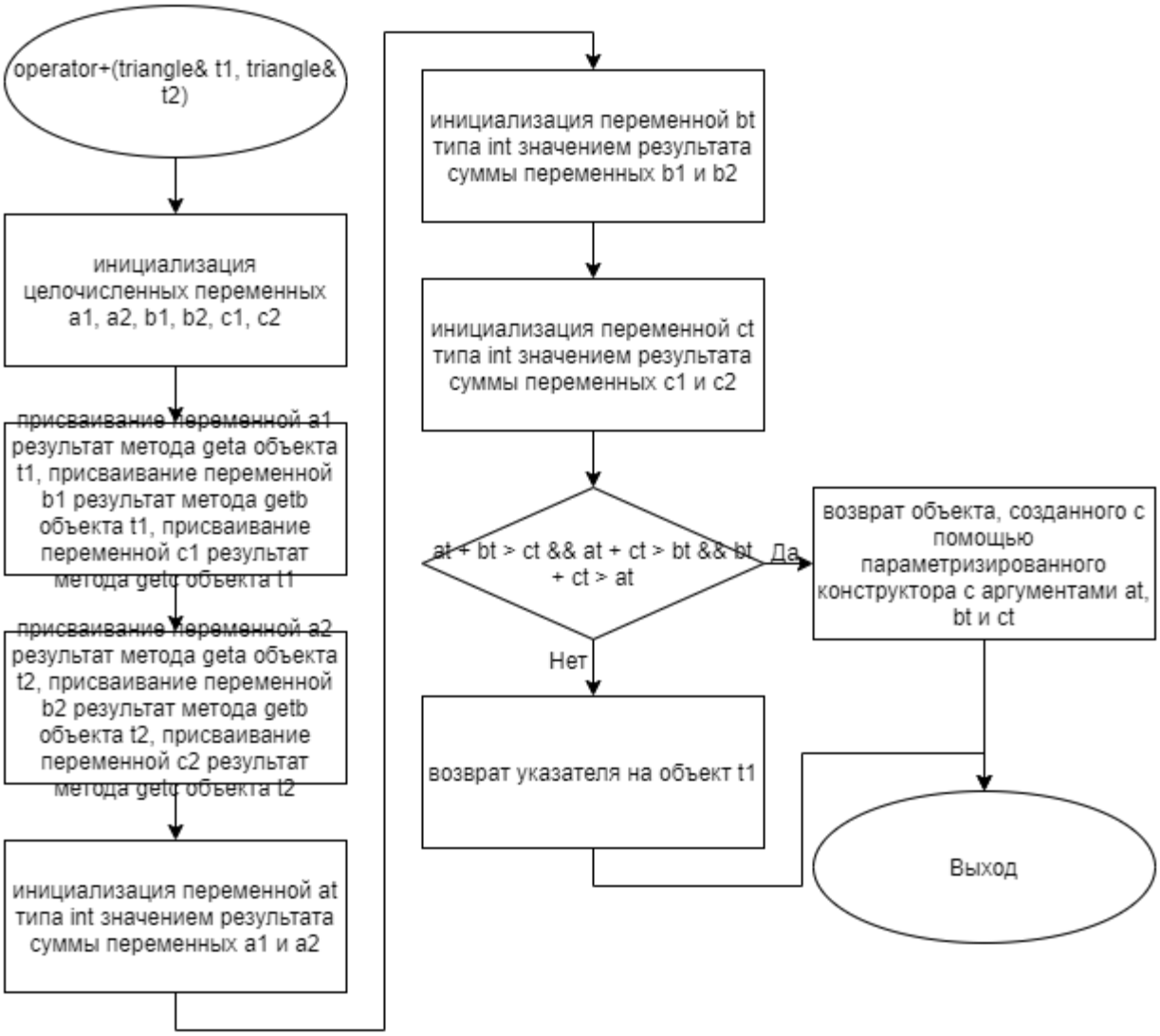

Рисунок 2 – Блок-схема алгоритма

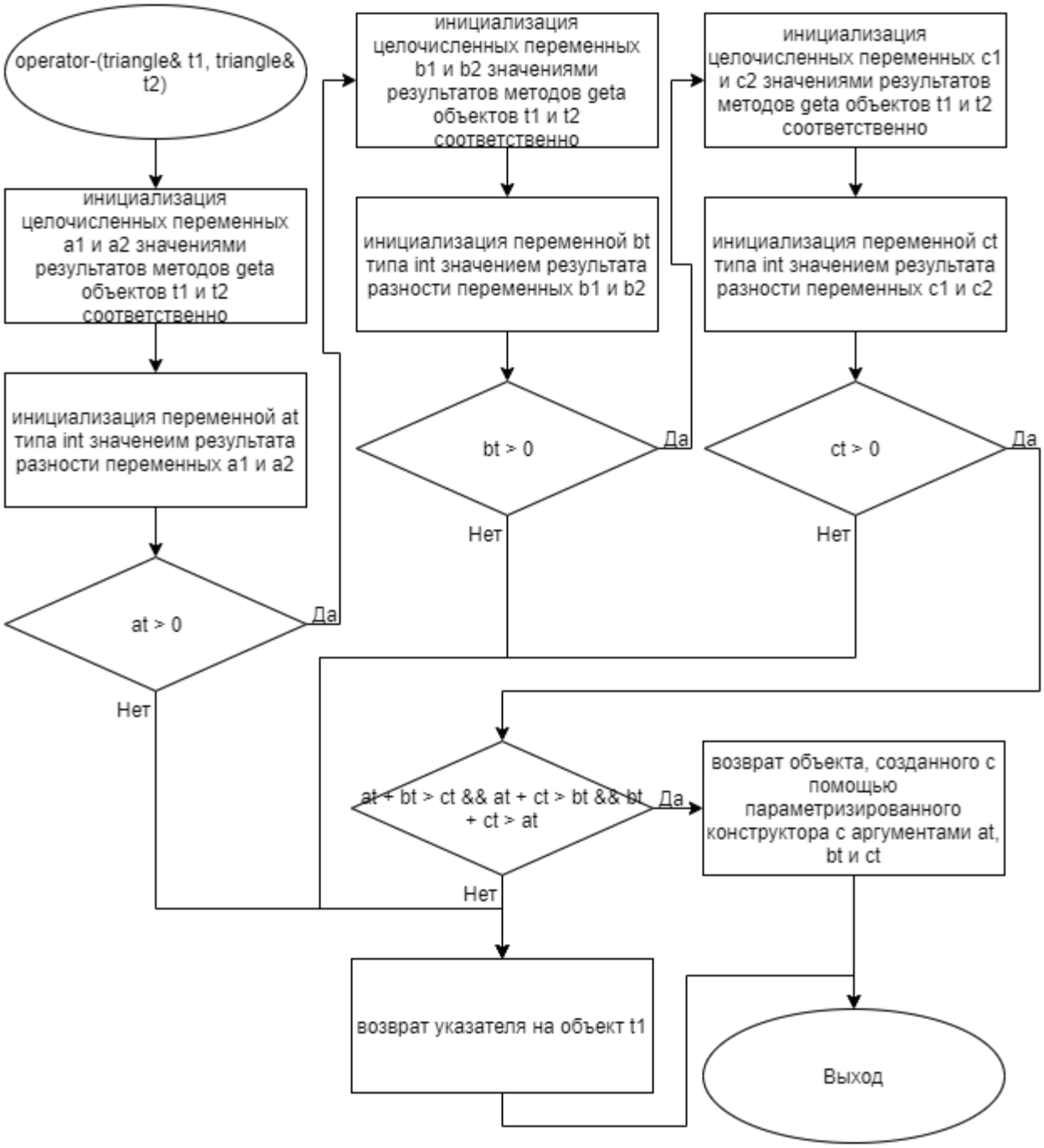

Рисунок 3 – Блок-схема алгоритма

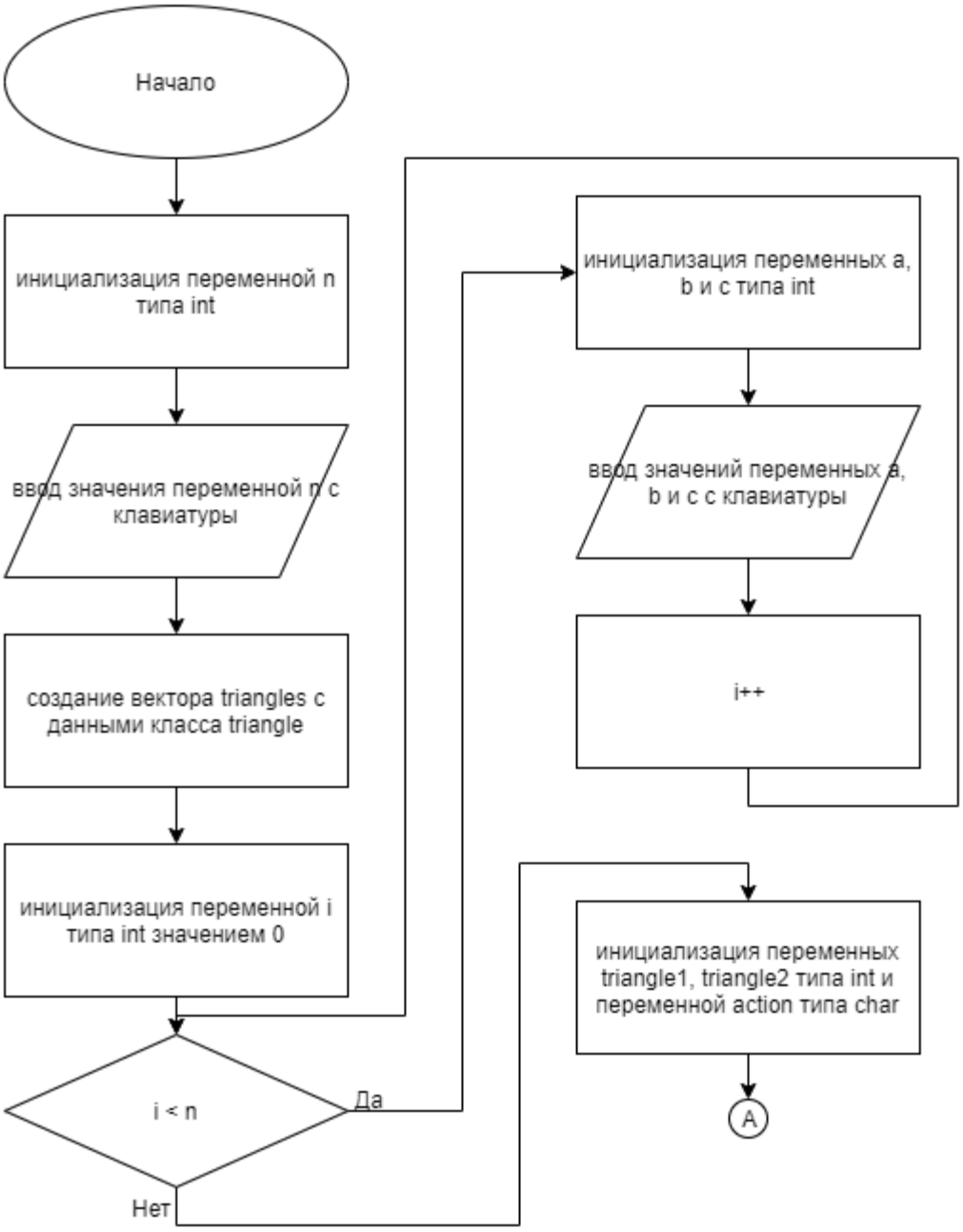

Рисунок 4 – Блок-схема алгоритма

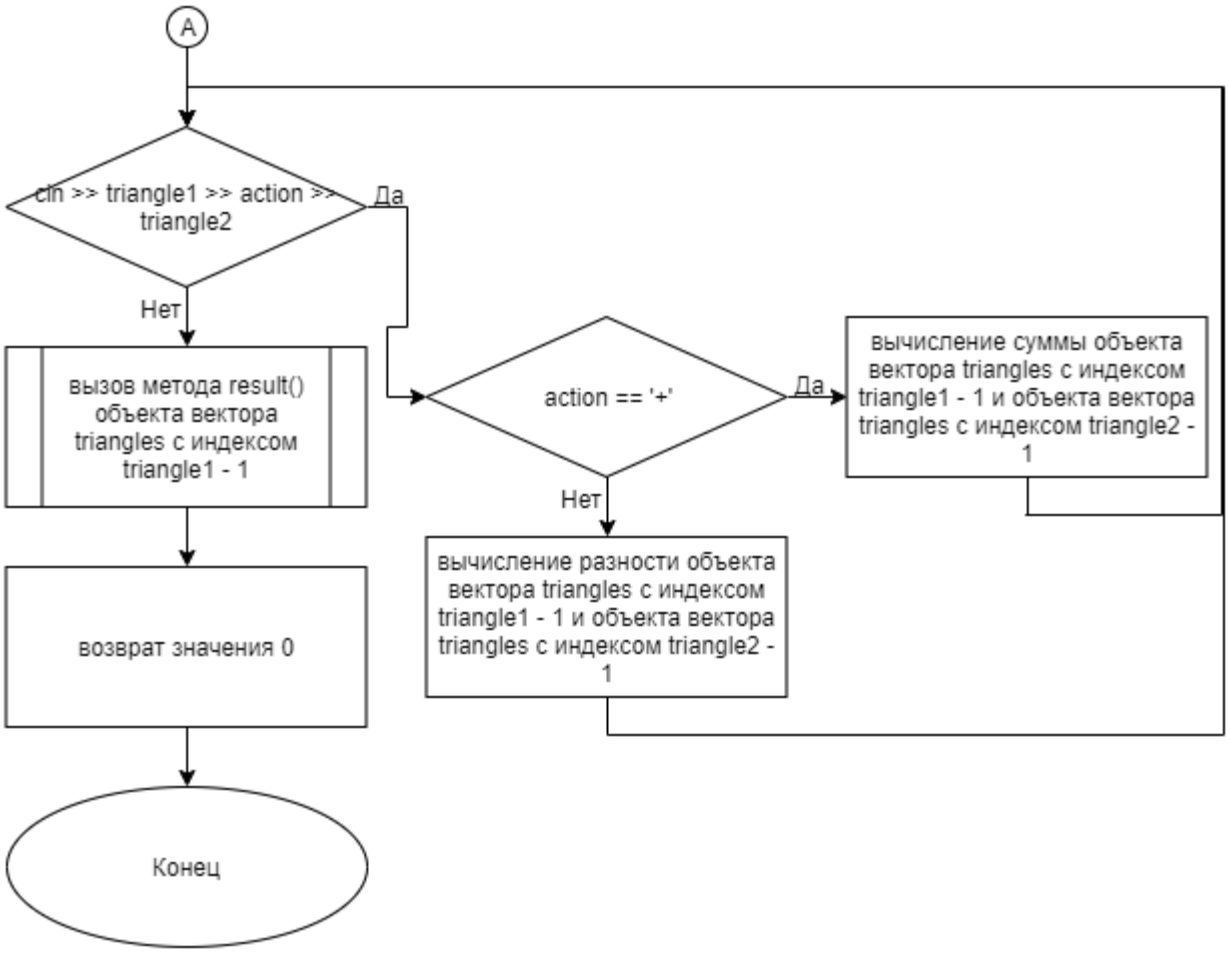

Рисунок 5 – Блок-схема алгоритма

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"

int main()
{
   int n;
   std::cin >> n;
   std::vector <triangle> triangles;
   for (int i = 0; i < n; ++i)
   {
      int a, b, c;
      std::cin >> a >> b >> c;
      triangles.push_back(triangle(a, b, c));
   }
   int triangle1, triangle2; char action;
   while (std::cin >> triangle1 >> action >> triangle2)
   {
      if (action == '+') triangles[triangle1 - 1] = triangles[triangle1 - 1]
+ triangles[triangle2 - 1];
      else  triangles[triangle1  -  1]  =  triangles[triangle1  -  1]  -
triangles[triangle2 - 1];
   }
   triangles[triangle1 - 1].result();
   return(0);
}
```

5.2 Файл triangle.cpp

Листинг 2 – triangle.cpp

```
#include "triangle.h"
```

```
#include <iostream>

triangle::triangle(int a, int b, int c)
{
   this -> a = a;
   this -> b = b;
   this -> c = c;
}
void triangle::result()
{ std::cout << "a = " << a << "; b = " << b << "; c = " << c << '.'; }
int triangle::geta()
{ return a; }
int triangle::getb()
{ return b; }
int triangle::getc()
{ return c; }

triangle operator+(triangle& t1, triangle& t2)
{
   int a1, a2, b1, b2, c1, c2;
   a1 = t1.geta(); b1 = t1.getb(); c1 = t1.getc();
   a2 = t2.geta(); b2 = t2.getb(); c2 = t2.getc();
   int at = a1 + a2;
   int bt = b1 + b2;
   int ct = c1 + c2;
   if (at + bt > ct && at + ct > bt && bt + ct > at)
      return triangle(at, bt, ct);
   else return t1;
}
triangle operator-(triangle& t1, triangle& t2)
{
   int a1 = t1.geta(), a2 = t2.geta();
   int at = a1 - a2;
   if (at > 0)
   {
      int b1 = t1.getb(), b2 = t2.getb();
      int bt = b1 - b2;
      if (bt > 0)
      {
         int c1 = t1.getc(), c2 = t2.getc();
         int ct = c1 - c2;
         if (ct > 0)
            if (at + bt > ct && at + ct > bt && bt + ct > at)
               return triangle(at, bt, ct);
      }
   }
   return t1;
}
```

5.3 Файл triangle.h

Листинг 3 – triangle.h

```
#ifndef __TRIANGLE__H
#define __TRIANGLE__H

class triangle
{
private:
   int a, b, c;
public:
   triangle(int a, int b, int c);
   void result();
   int geta();
   int getb();
   int getc();
};

triangle operator+(triangle& t1, triangle& t2);
triangle operator-(triangle& t1, triangle& t2);

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 9.

Таблица 9 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
4 2 3 4 10 11 12 6 3 7 12 31 40 2 - 3 1 + 2	a = 6; b = 11; c = 9.	a = 6; b = 11; c = 9.
3 10 11 12 23 33 43 10 20 30 1 - 3 2 + 1	a = 33; b = 44; c = 55.	a = 33; b = 44; c = 55.
2 12 15 19 3 4 5 1 - 2 2 - 1	a = 3; b = 4; c = 5.	a = 3; b = 4; c = 5.
2 12 13 14 3 4 5 1 + 2	a = 15; b = 17; c = 19.	a = 15; b = 17; c = 19.

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-avrora.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-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

4. Шилдт Г. C++: 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).