

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 функции main.....	8
3.2 Алгоритм метода print класса Triangle.....	8
3.3 Алгоритм функции operator.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	10
5 КОД ПРОГРАММЫ.....	12
5.1 Файл main.cpp.....	12
5.2 Файл Triangle.cpp.....	13
5.3 Файл Triangle.h.....	14
6 ТЕСТИРОВАНИЕ.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	16

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:

- пространство имен `cin`, `cout`;
- `if...else`.

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 функции main

Функционал: запуск программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменных n, a, b, c типа int	2
2		ввод значений переменной n создание пустого массива	3
3	i < n		4
			7
4		ввод значений переменных a, b	5
5		создание объекта obj класса triangles с параметрами a, b, c	6
6		добавление объекта obj в массиве	7
7		объявление переменных	Ø

3.2 Алгоритм метода print класса Triangle

Функционал: вывод значений.

Параметры: нет.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода *print* класса *Triangle*

№	Предикат	Действия	№ перехода
1		вывод на экран сообщения	Ø

3.3 Алгоритм функции *operator*

Функционал: оператор вычитания.

Параметры: нет.

Возвращаемое значение: *triangle*.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции *operator*

№	Предикат	Действия	№ перехода
1		возвращение значения объекта 11	2
2			Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

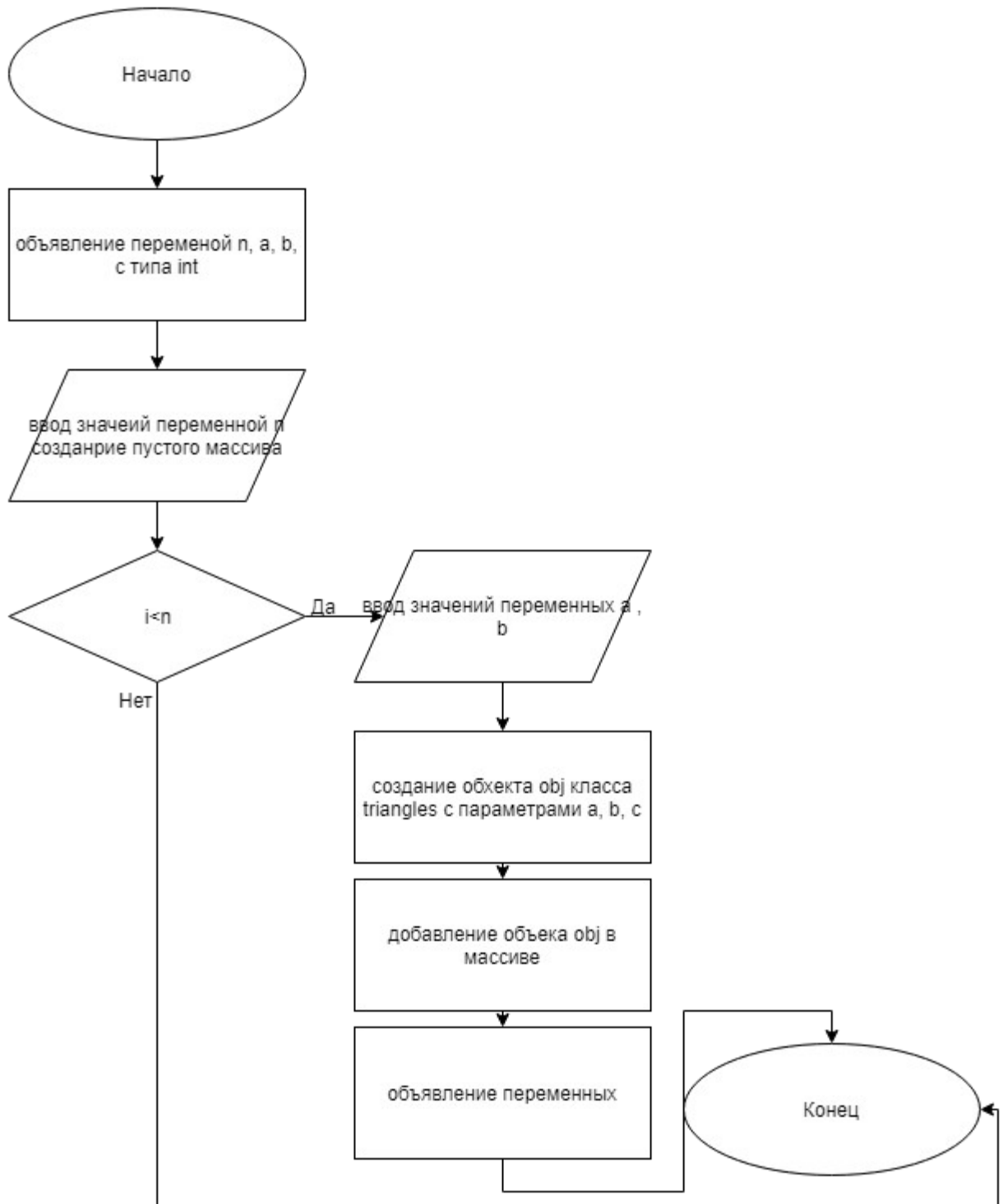


Рисунок 1 – Блок-схема алгоритма

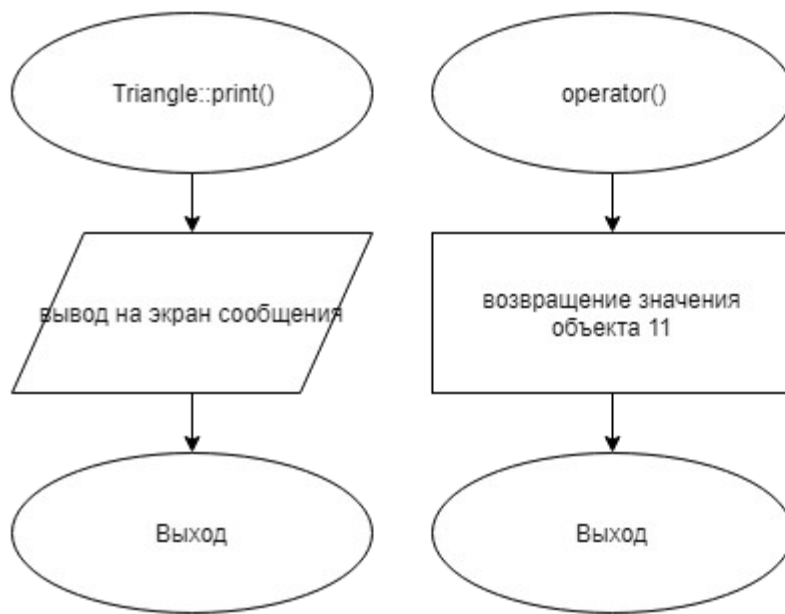


Рисунок 2 – Блок-схема алгоритма

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 "Triangle.h"
#include <vector>
using namespace std;

Triangle operator + (Triangle t1, Triangle t2)
{
    int a = t1.a + t2.a;
    int b = t1.b + t2.b;
    int c = t1.c + t2.c;
    if ((a + b > c) & (a + c > b) & (b + c > a))
    {
        t1.a = a;
        t1.b = b;
        t1.c = c;
    }
    return t1;
}

Triangle operator - (Triangle t1, Triangle t2)
{
    int a = t1.a - t2.a;
    int b = t1.b - t2.b;
    int c = t1.c - t2.c;
    if ((a + b > c) & (a + c > b) & (b + c > a))
    {
        t1.a = a;
        t1.b = b;
        t1.c = c;
    }
    return t1;
}

int main()
{
    int a, b, c, n;
    cin >> n;
```

```

vector <Triangle> triangles;
for (int i = 0; i < n; i++)
{
    cin >> a >> b >> c;
    Triangle obj(a, b, c);
    triangles.push_back(obj);
}
int trian1, trian2;
char operation;
while (cin >> trian1, cin >> operation, cin >> trian2)
{
    if (operation == '+')
    {
        triangles[trian1 - 1] = triangles[trian1 - 1] + triangles[trian2 -
1];
    }
    else if (operation == '-')
    {
        triangles[trian1 - 1] = triangles[trian1 - 1] - triangles[trian2 -
1];
    }
}
triangles[trian1 - 1].print();
return(0);
}

```

5.2 Файл Triangle.cpp

Листинг 2 – Triangle.cpp

```

#include "Triangle.h"
#include <cmath>
Triangle::Triangle(int a, int b, int c)
{
    this->a = a;
    this->b = b;
    this->c = c;
}
void Triangle :: print()
{
    cout << "a = " << a << "; " << "b = " << b << "; "<< "c = " << c << "." <<
endl;
}

```

5.3 Файл Triangle.h

Листинг 3 – Triangle.h

```
#ifndef __TRIANGLE__H
#define __TRIANGLE__H
#include <iostream>
using namespace std;
class Triangle
{
public:
    int a, b, c;
    Triangle(int a, int b, int c);
    void print();
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 3 4 5 5 12 13 1 + 2	a = 8; b = 16; c = 18.	a = 8; b = 16; c = 18.

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).