

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	5
1.2 Описание выходных данных.....	5
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	8
3.1 Алгоритм функции main.....	8
3.2 Алгоритм конструктора класса Triangle.....	8
3.3 Алгоритм метода calculate_perimeter класса Triangle.....	9
3.4 Алгоритм метода calculate_square класса Triangle.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	10
5 КОД ПРОГРАММЫ.....	12
5.1 Файл main.cpp.....	12
5.2 Файл Triangle.cpp.....	12
5.3 Файл Triangle.h.....	13
6 ТЕСТИРОВАНИЕ.....	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	15

1 ПОСТАНОВКА ЗАДАЧИ

Создать объект «треугольник», который содержит длины сторон треугольника.

Значения длин сторон натуральные числа.

Объект вычисляет периметр и площадь треугольника.

Функционал:

- параметризированный конструктор с параметрами длин сторон;
- метод вычисления и возврата значения периметра;
- метод вычисления и возврата значения площади.

Написать программу:

1. Вводит стороны треугольника.
2. Создает объект «треугольник»,
3. Выводит периметр.
4. Выводит площадь.

1.1 Описание входных данных

Три целых числа, соответствующие длинам сторон треугольника, разделенные пробелом.

Подразумевается, что для заданных данных треугольник существует.

1.2 Описание выходных данных

Первая строка:

P = «периметр»

Вторая строка:

$S = \text{«площадь»}$

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект cin класса istream;
- объект cout класса ostream;
- объект класса Triangle;
- функция sqrt для извлечение квадратного корня.

Класс Triangle:

- свойства/поля:
 - поле сторона треугольника:
 - наименование — a_;
 - тип — int;
 - модификатор доступа — private;
 - поле сторона треугольника:
 - наименование — b_;
 - тип — int;
 - модификатор доступа — private;
 - поле сторона треугольника:
 - наименование — c_;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод calculate_square — вычисление площади треугольника;
 - метод calculate_perimeter — вычисление периметра треугольника.

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: точка входа в программу.

Параметры: нет.

Возвращаемое значение: целочисленное - индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление переменных a,b,c	2
2		Ввод значений переменных a,b,c с клавиатуры	3
3		Объявление объекта triangle класса Traingle	4
4		Вывод "P = "; вызов метода calculate_perimeter();	5
5		Вывод "S = "; вызов метода calculate_square()	Ø

3.2 Алгоритм конструктора класса Triangle

Функционал: параметризованный конструктор.

Параметры: целочисленные a,b,c - стороны треугольника.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса *Triangle*

№	Предикат	Действия	№ перехода
1		присваивание значений параметров a, b, c к закрытым свойствам	Ø

3.3 Алгоритм метода *calculate_perimeter* класса *Triangle*

Функционал: вычисление периметра треугольника.

Параметры: нет.

Возвращаемое значение: вещественное число двойной точности.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *calculate_perimeter* класса *Triangle*

№	Предикат	Действия	№ перехода
1		Возврат $a_ + b_ + c_$	Ø

3.4 Алгоритм метода *calculate_square* класса *Triangle*

Функционал: вычисление площади треугольника.

Параметры: нет.

Возвращаемое значение: вещественное число двойной точности.

Алгоритм метода представлен в таблице 4.

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

№	Предикат	Действия	№ перехода
1		Инициализация вещественной переменной двойной точности $p = (a+b+c) / 2$	2
2		Инициализация вещественной переменной двойной точности $S =$ рассчитаной через полупериметр площади треугольника	3
3		Возврат S	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

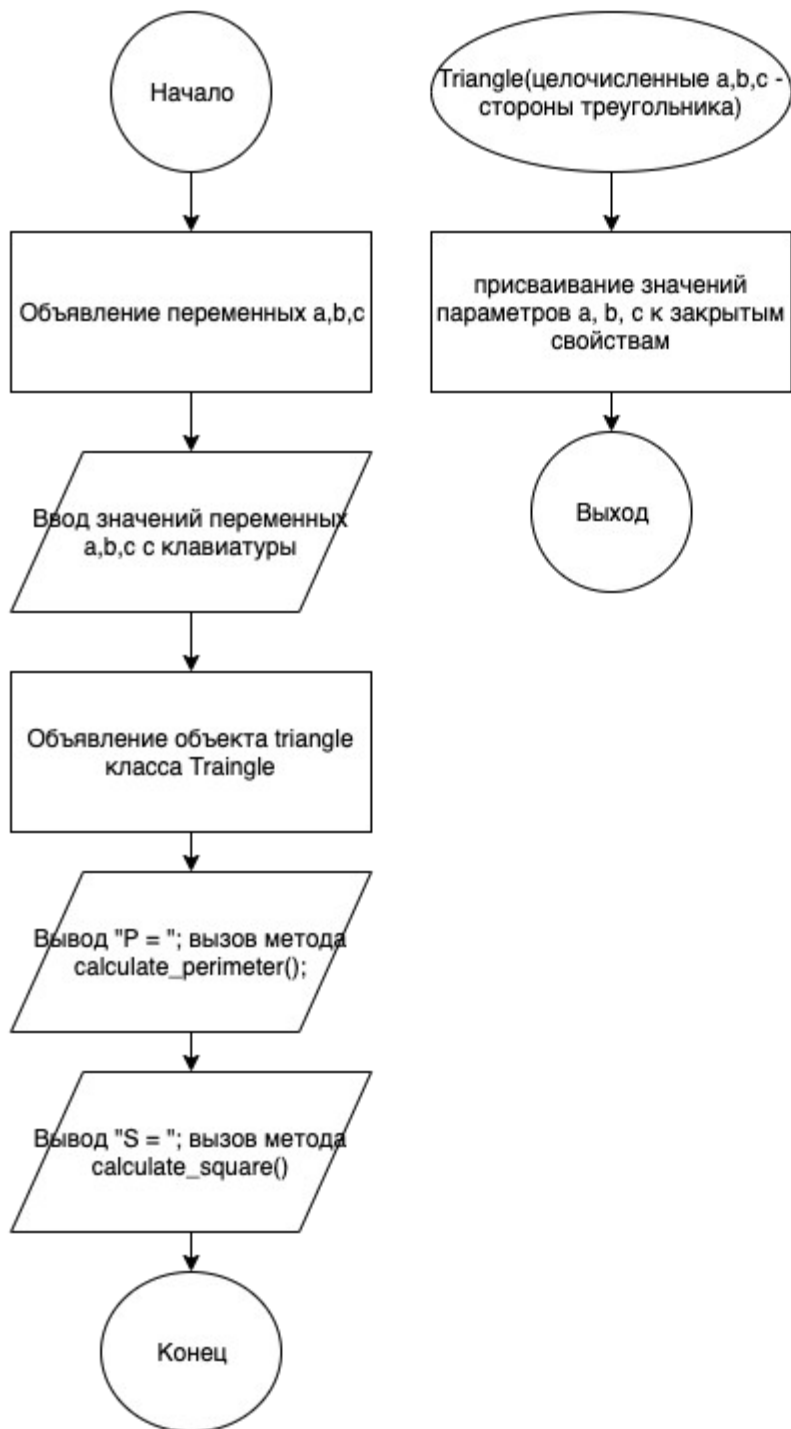


Рисунок 1 – Блок-схема алгоритма

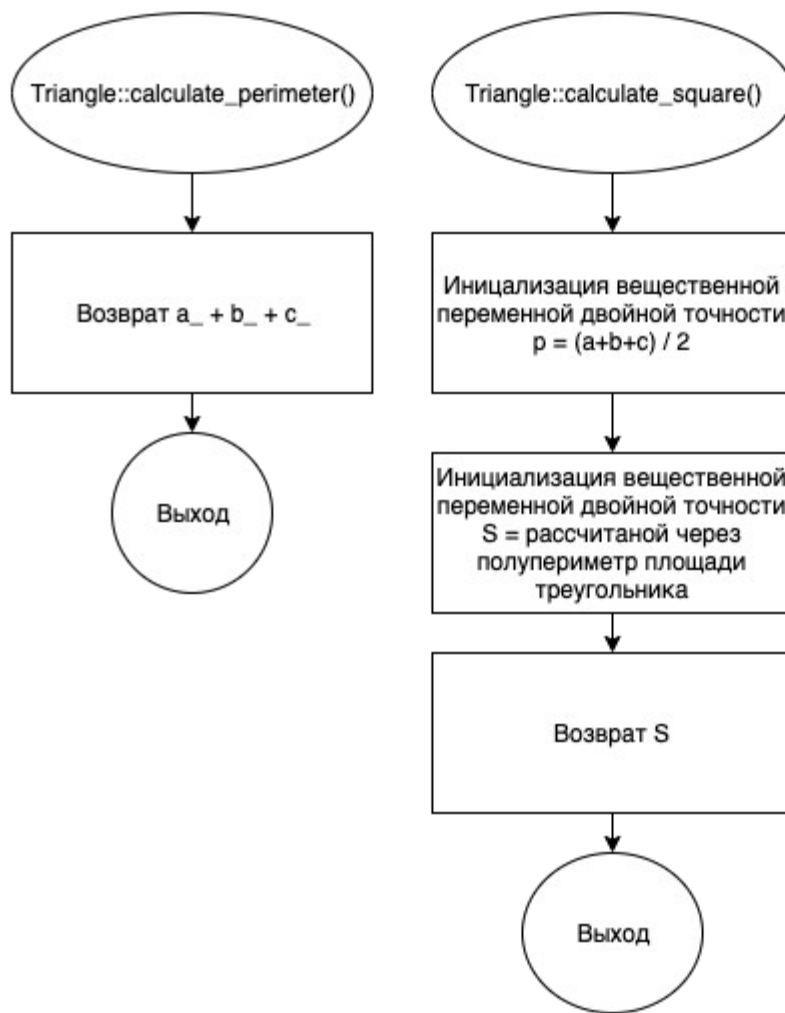


Рисунок 2 – Блок-схема алгоритма

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"

int main()
{
    int a, b, c;

    std::cin >> a >> b >> c;

    Triangle triangle(a,b,c);

    std::cout << "P = " << triangle.calculate_perimeter() << std::endl;
    std::cout << "S = " << triangle.calculate_square();
    return 0;
}
```

5.2 Файл Triangle.cpp

Листинг 2 – Triangle.cpp

```
#include "Triangle.h"
#include <cmath>
#include <iostream>

Triangle::Triangle(double a, double b, double c) {
    a_ = a;
    b_ = b;
    c_ = c;
}

double Triangle::calculate_perimeter() {
```

```

        return a_ + b_ + c_;
    }

    double Triangle::calculate_square() {
        double p = calculate_perimeter() / 2;
        double S = std::sqrt(p*(p - a_)*(p - b_)*(p - c_));

        return S;
    }

```

5.3 Файл Triangle.h

Листинг 3 – Triangle.h

```

#ifndef ZANASHIH_TRIANGLE_H
#define ZANASHIH_TRIANGLE_H

class Triangle {
private:
    int a_, b_, c_;

public:
    Triangle(double a, double b, double c);

    double calculate_perimeter();
    double calculate_square();

};

#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
3 4 5	$P = 12$ $S = 6$	$P = 12$ $S = 6$
1 2 3	$P = 6$ $S = 0$	$P = 6$ $S = 0$

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).