

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.....	9
3.1 Алгоритм конструктора класса triangle.....	9
3.2 Алгоритм метода Р класса triangle.....	9
3.3 Алгоритм метода S класса triangle.....	10
3.4 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
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 `triangle` класса `triangle` предназначен для объект класса "треугольник";
- функция `main` для основная функция программы;
- функция `sqrt` для функция для извлечения квадратного корня из числа, описанная в библиотеке `<cmath>`;
- `cin` - объект стандартного потока ввода с клавиатуры;
- `cout` - объект стандартного потока вывода на экран.

Класс `triangle`:

- свойства/поля:
  - поле первая сторона треугольника:
    - наименование — `a`;
    - тип — `int`;
    - модификатор доступа — `private`;
  - поле вторая сторона треугольника:
    - наименование — `b`;
    - тип — `int`;
    - модификатор доступа — `private`;
  - поле третья сторона треугольника:
    - наименование — `c`;
    - тип — `int`;
    - модификатор доступа — `private`;
- функционал:
  - метод `triangle` — параметризованный конструктор;
  - метод `P` — вычисляет периметр объекта;

- о метод  $S$  — вычисляет площадь 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: параметризованный 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 объекта a значение параметра a	2
2		присваивание полю объекта b значение параметра b	3
3		присваивание полю объекта c значение параметра c	Ø

### 3.2 Алгоритм метода *P* класса *triangle*

Функционал: вычисляет периметр объекта.

Параметры: нет.

Возвращаемое значение: double.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода *P* класса *triangle*

№	Предикат	Действия	№ перехода
1		возвращает значение суммы полей объекта a, b и c	Ø

### 3.3 Алгоритм метода S класса triangle

Функционал: вычисляет площадь объекта.

Параметры: нет.

Возвращаемое значение: double.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода S класса triangle

№	Предикат	Действия	№ перехода
1		инициализация переменной p значением результата метода P данного объекта	2
2		возврат значения результата функции sqrt от произведения p, p-a, p-b и p-c	Ø

### 3.4 Алгоритм функции main

Функционал: основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		создание переменных a, b и c типа int	2
2		ввод значений переменных a, b и c с клавиатуры	3
3		создание объекта triangle класса triangle с помощью параметризованного конструктора с аргументами a, b и c	4
4		вывод на экран "P = " и результата метода P() объекта triangle	5
5		вывод на экран "\nS = " и результата метода S() объекта triangle	6
6		возврат значения 0	Ø



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

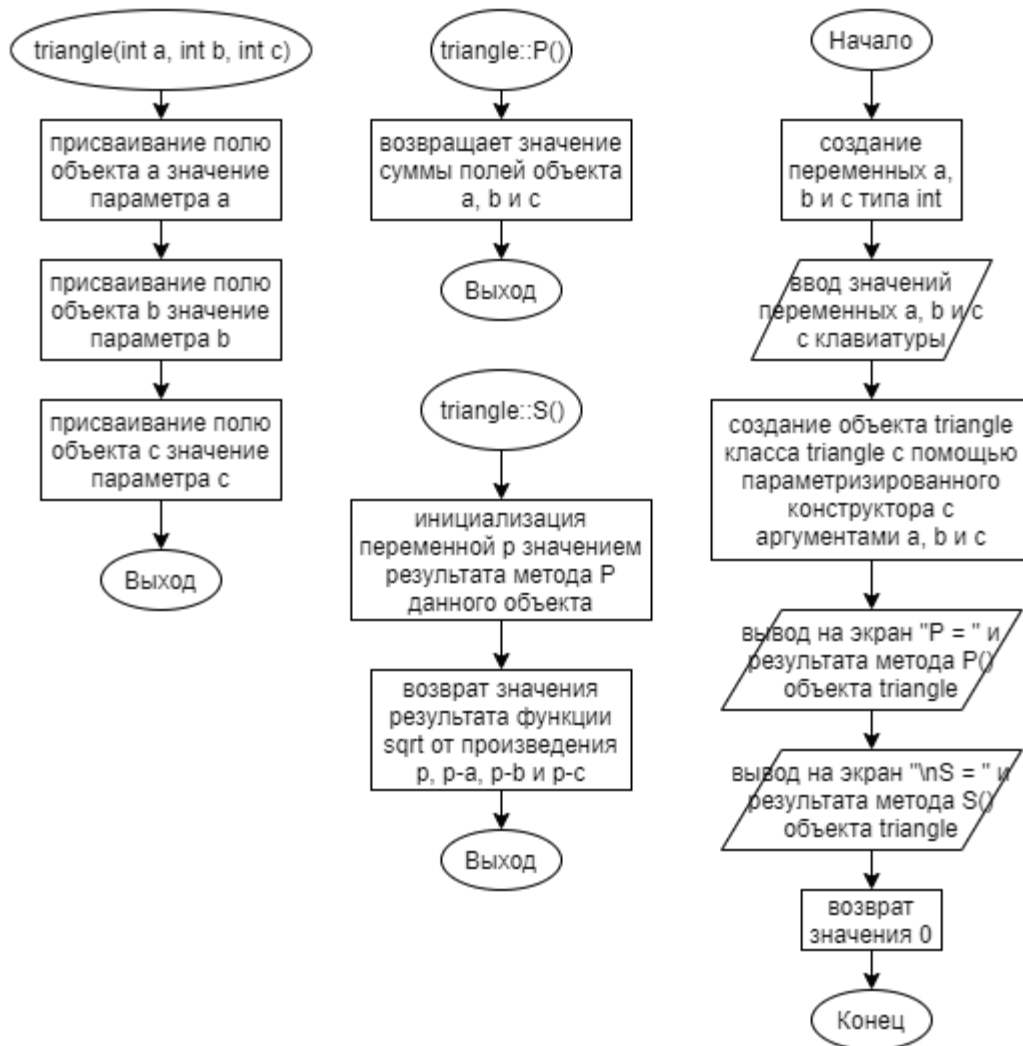


Рисунок 1 – Блок-схема алгоритма

## 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.P();
    std::cout << "\nS = " << triangle.S();
    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;
}
double triangle::P()
{
    return a + b + c;
}
double triangle::S()
{

```

```
        double p = P()/2;  
        return sqrt(p * (p - a) * (p - b) * (p - c));  
    }
```

## 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);  
    double P();  
    double S();  
};  
  
#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
5 8 10	P = 23 S = 19.81	P = 23 S = 19.81
23 40 21	P = 84 S = 183.074	P = 84 S = 183.074
78 123 63	P = 264 S = 2103.92	P = 264 S = 2103.92

## 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](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](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).