

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

# 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 Хранение и работа со сторонами треугольника;
- функция main для Главная функция программы;
- библиотека iostream;
- библиотека cmath;
- пространство имён std;
- Объект стандартного потока ввода cin с клавиатуры;
- Объект стандартного потока вывода cout на экран.

Класс Triangle:

- свойства/поля:
  - поле Хранение значения 1-ой стороны треугольника:
    - наименование — side\_a;
    - тип — int;
    - модификатор доступа — protected;
  - поле Хранение значения 2-ой стороны треугольника:
    - наименование — side\_b;
    - тип — int;
    - модификатор доступа — protected;
  - поле Хранение значения 3-ей стороны треугольника:
    - наименование — side\_c;
    - тип — int;
    - модификатор доступа — protected;
- функционал:
  - метод Triangle — параметризированный конструктор,

присваивающий скрытым полям `side_a`, `side_b`, `side_c` переданные в него значения посредством указателя `this`;

- о метод `Perimetr` — вычисление периметра треугольника;
- о метод `Area` — вычисление площади 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 конструктора класса **Triangle**

Функционал: параметризированный конструктор, присваивающий скрытым полям `side_a`, `side_b`, `side_c` переданные в него значения посредством указателя `this`.

Параметры: целое, `side_a`, `side_b`, `side_c`, заданные значения сторон треугольника.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса *Triangle*

№	Предикат	Действия	№ перехода
1		Присвоение скрытому полю <code>side_a</code> переданного значения посредством указателя <code>this</code>	2
2		Присвоение скрытому полю <code>side_b</code> переданного значения посредством указателя <code>this</code>	3
3		Присвоение скрытому полю <code>side_c</code> переданного значения посредством указателя <code>this</code>	Ø

### 3.2 Алгоритм метода **Perimetr** класса **Triangle**

Функционал: вычисление периметра треугольника.

Параметры: нет.

Возвращаемое значение: целое, периметр треугольника.

Алгоритм метода представлен в таблице 2.

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

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной <i>perimetr</i> значением суммы полей <i>side_a</i> , <i>side_b</i> и <i>side_c</i>	2
2		Возврат значения переменной <i>perimetr</i>	∅

### 3.3 Алгоритм метода *Area* класса *Triangle*

Функционал: вычисление площади треугольника.

Параметры: нет.

Возвращаемое значение: с плавающей запятой, площадь треугольника.

Алгоритм метода представлен в таблице 3.

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

№	Предикат	Действия	№ перехода
1		Инициализация переменной <i>half_perimetr</i> с плавающей точкой значением половины суммы полей <i>side_a</i> , <i>side_b</i> и <i>side_c</i>	2
2		Инициализация переменной <i>area</i> с плавающей точкой следующим значением: $\sqrt{\text{half\_perimetr} * (\text{half\_perimetr} - \text{side\_a}) * (\text{half\_perimetr} - \text{side\_b}) * (\text{half\_perimetr} - \text{side\_c})}$	3
3		Возврат значения переменной <i>area</i>	∅

### 3.4 Алгоритм функции *main*

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 4.



Таблица 4 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление целочисленных переменных <code>side_a</code> , <code>side_b</code> и <code>side_c</code>	2
2		Ввод значений переменных <code>side_a</code> , <code>side_b</code> и <code>side_c</code>	3
3		Создание объекта <code>Triangle</code> класса <code>Triangle</code> с передачей в 4 параметризованный конструктор значений переменных <code>side_a</code> , <code>side_b</code> и <code>side_c</code>	
4		Вывод: "P = ", <<возвращаемое значение метода <code>Perimetr()</code> объекта <code>Triangle</code> >>	5
5		Переход на новую строку	6
6		Вывод: "S = ", <<возвращаемое значение метода <code>Area()</code> объекта <code>Triangle</code> >>	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

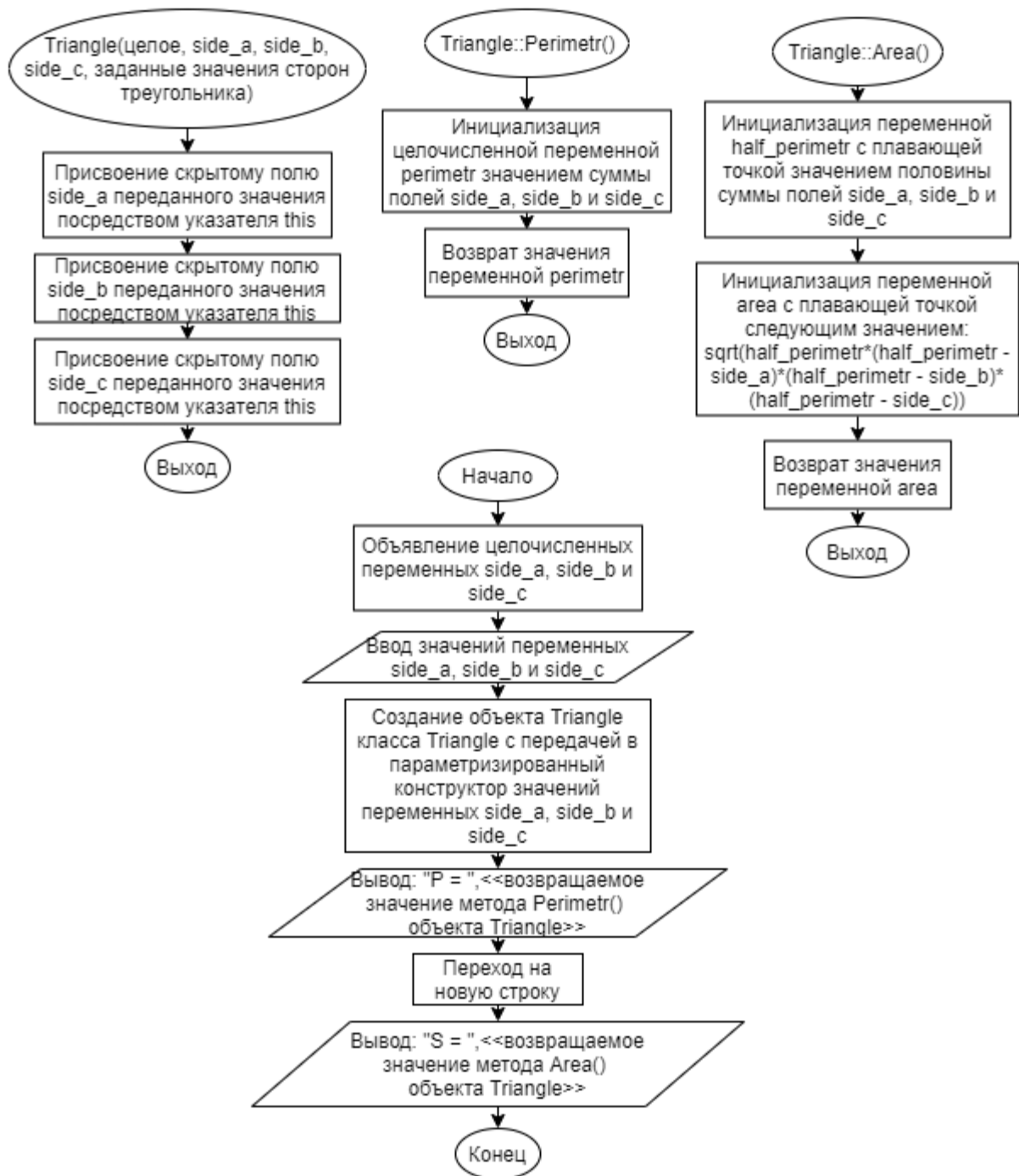


Рисунок 1 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл main.cpp

*Листинг 1 – main.cpp*

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

int main()
{
    int side_a, side_b, side_c;
    cin >> side_a >> side_b >> side_c;
    Triangle Triangle(side_a, side_b, side_c);
    cout << "P = " << Triangle.Perimetr();
    cout << "\nS = " << Triangle.Area();
    return 0;
}
```

### 5.2 Файл Triangle.cpp

*Листинг 2 – Triangle.cpp*

```
#include "Triangle.h"
#include <cmath>
Triangle :: Triangle(int side_a, int side_b, int side_c)
{
    this -> side_a = side_a;
    this -> side_b = side_b;
    this -> side_c = side_c;
}
float Triangle :: Area()
{
    float half_perimetr = (side_a + side_b + side_c)/2.0;
    float area = sqrt(half_perimetr*(half_perimetr - side_a)*(half_perimetr - side_b)*(half_perimetr - side_c));
    return area;
}
int Triangle :: Perimetr()
```

```
{  
    int perimetr = side_a + side_b + side_c;  
    return perimetr;  
}
```

## 5.3 Файл Triangle.h

*Листинг 3 – Triangle.h*

```
#ifndef __DANCINGYELLOWTRIANGLE__H  
#define __DANCINGYELLOWTRIANGLE__H  
class Triangle  
{  
    public:  
    Triangle(int side_a, int side_b, int side_c);  
    float Area();  
    int Perimetr();  
    private:  
    int side_a;  
    int side_b;  
    int side_c;  
};  
  
#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
6 6 6	P = 18 S = 15.5885	P = 18 S = 15.5885
7 8 9	P = 24 S = 26.8328	P = 24 S = 26.8328
14 25 36	P = 75 S = 128.544	P = 75 S = 128.544

## 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).