

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 метода S класса triangle..... | 8 |
| 3.2 Алгоритм метода Р класса triangle..... | 8 |
| 3.3 Алгоритм конструктора класса triangle..... | 9 |
| 3.4 Алгоритм функции main..... | 9 |
| 4 БЛОК-СХЕМЫ АЛГОРИТМОВ..... | 10 |
| 5 КОД ПРОГРАММЫ..... | 11 |
| 5.1 Файл main.cpp..... | 11 |
| 5.2 Файл triangle.cpp..... | 11 |
| 5.3 Файл triangle.h..... | 12 |
| 6 ТЕСТИРОВАНИЕ..... | 13 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 14 |

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 `t` класса `triangle` предназначен для Объект треугольника для вычисления периметра и площади;
- `cin/cout` - объекты стандартного потока ввода/вывода;
- `sqrt` - функция вычисления квадратного корня из числа.

Класс `triangle`:

- свойства/поля:
 - поле Сторона 1:
 - наименование — `a`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле Сторона 2:
 - наименование — `b`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле Сторона 3:
 - наименование — `b`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - метод `P` — Вычисление и возврат периметра треугольника;
 - метод `S` — Вычисление и возврат площади треугольника;
 - метод `triangle` — Параметризированный конструктор.

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 метода S класса triangle

Функционал: Вычисление и возврат площади треугольника.

Параметры: нет.

Возвращаемое значение: float - площадь треугольника.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода S класса triangle

| № | Предикат | Действия | № перехода |
|---|----------|----------------------------------------------------------------------------|---------------|
| 1 | | Инициализация переменной p типа float значением вызова метода P() * 0.5 | 2 |
| 2 | | Возврат $\sqrt{p * (p - a) * (p - b) * (p - c)}$ | Ø |

3.2 Алгоритм метода P класса triangle

Функционал: Вычисление и возврат периметра треугольника.

Параметры: нет.

Возвращаемое значение: int - периметр треугольника.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода P класса triangle

| № | Предикат | Действия | № перехода |
|---|----------|---------------------|---------------|
| 1 | | Возврат $a + b + c$ | Ø |

3.3 Алгоритм конструктора класса triangle

Функционал: Параметризированный конструктор.

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

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса triangle

| № | Предикат | Действия | № перехода |
|---|----------|-------------------------------------------------|---------------|
| 1 | | Установка значения поля a значением параметра a | 2 |
| 2 | | Установка значения поля b значением параметра b | 3 |
| 3 | | Установка значения поля c значением параметра c | Ø |

3.4 Алгоритм функции main

Функционал: Главная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции main

| № | Предикат | Действия | № перехода |
|---|----------|----------------------------------------------------------------------------------------------------------|---------------|
| 1 | | Инициализация переменных a, b, c типа Int | 2 |
| 2 | | Ввод значений a, b, c | 3 |
| 3 | | Инициализация объекта t класса triangle с передачей параметров a, b, c параметризированному конструктору | 4 |
| 4 | | Вывод "P = " значение вызова метода P у объекта t | 5 |
| 5 | | Вывод "S = " значение вызова метода S у объекта t | Ø |

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

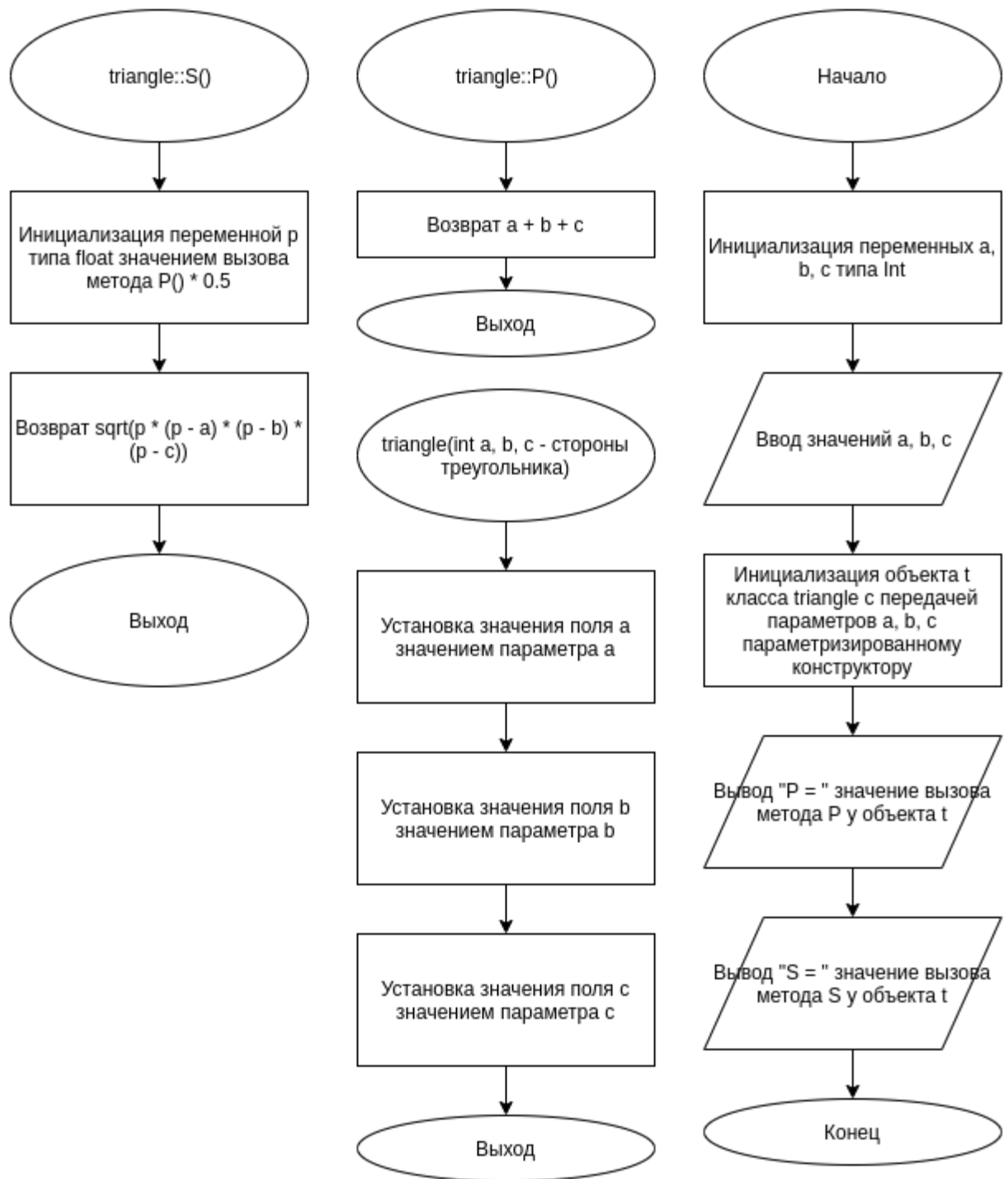


Рисунок 1 – Блок-схема алгоритма

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 "triangle.h"

int main()
{
    int a, b, c;
    cin >> a >> b >> c;
    triangle t(a, b, c);
    cout << "P = " << t.P() << endl;
    cout << "S = " << t.S() << endl;
    return(0);
}
```

5.2 Файл triangle.cpp

Листинг 2 – triangle.cpp

```
#include "triangle.h"

triangle::triangle(int a, int b, int c) {
    this->a = a;
    this->b = b;
    this->c = c;
}

float triangle::S() {
    float p = P() * 0.5;
    return sqrt(p * (p - a) * (p - b) * (p - c));
}

float triangle::P() {
```

```
    return a + b + c;  
}
```

5.3 Файл triangle.h

Листинг 3 – triangle.h

```
#ifndef __TRIANGLE__H  
#define __TRIANGLE__H  
  
#include <cmath>  
#include <iostream>  
using namespace std;  
  
class triangle  
{  
private:  
    int a, b, c;  
public:  
    triangle(int, int, int);  
    float P();  
    float 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 |

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).