

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 Р класса treangle.....	8
3.2 Алгоритм метода S класса treangle.....	8
3.3 Алгоритм метода параметризированный конструктор класса treangle.....	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 treangle.cpp.....	11
5.3 Файл treangle.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 `obj` класса `treangle` предназначен для вычисление периметра и площади треугольника;
- `cin` - объект стандартного потока ввода;
- `cout` - объект стандартного потока вывода.

Класс `treangle`:

- свойства/поля:
 - поле 1 сторона треугольника:
 - наименование — `a`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле 2 сторона треугольника:
 - наименование — `b`;
 - тип — `int`;
 - модификатор доступа — `private`;
 - поле 3 сторона треугольника:
 - наименование — `c`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - метод `P` — возвращает значение периметра треугольника;
 - метод `S` — возвращает значение площади треугольника;
 - метод параметризованный конструктор — параметризованный конструктор с параметрами длин сторон.

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 P класса *treangle*

Функционал: возвращает значение периметра треугольника.

Параметры: none.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода P класса *treangle*

№	Предикат	Действия	№ перехода
1		возвращение значение периметра треугольника	Ø

3.2 Алгоритм метода S класса *treangle*

Функционал: возвращает значение площади треугольника.

Параметры: none.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода S класса *treangle*

№	Предикат	Действия	№ перехода
1		возвращение значение площади треугольника	Ø

3.3 Алгоритм метода параметризированный конструктор класса `triangle`

Функционал: параметризированный конструктор с параметрами длин сторон.

Параметры: `int a`, `int b`, `int c`.

Возвращаемое значение: `none`.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода параметризированный конструктор класса `triangle`

№	Предикат	Действия	№ перехода
1		передает закрытому свойству <code>a</code> значение параметра <code>a</code>	2
2		передает закрытому свойству <code>b</code> значение параметра <code>b</code>	3
3		передает закрытому свойству <code>c</code> значение параметра <code>c</code>	Ø

3.4 Алгоритм функции `main`

Функционал: запуск программы.

Параметры: нет.

Возвращаемое значение: `int` - код ошибки.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции `main`

№	Предикат	Действия	№ перехода
1		объявление переменных <code>a</code> , <code>b</code> , <code>c</code> типа <code>int</code>	2
2		ввод значений переменных <code>a</code> , <code>b</code> , <code>c</code> с клавиатуры	3
3		создание объекта <code>obj</code> класса <code>triangle</code> с параметрами <code>a</code> , <code>b</code> , <code>c</code>	4
4		вывод на экран " <code>P = (возвращаемое значение P у объекта obj)</code> "	5
5		вывод на экран " <code>S = (возвращаемое значение S у объекта obj)</code> "	6
6		возврат значения 0	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

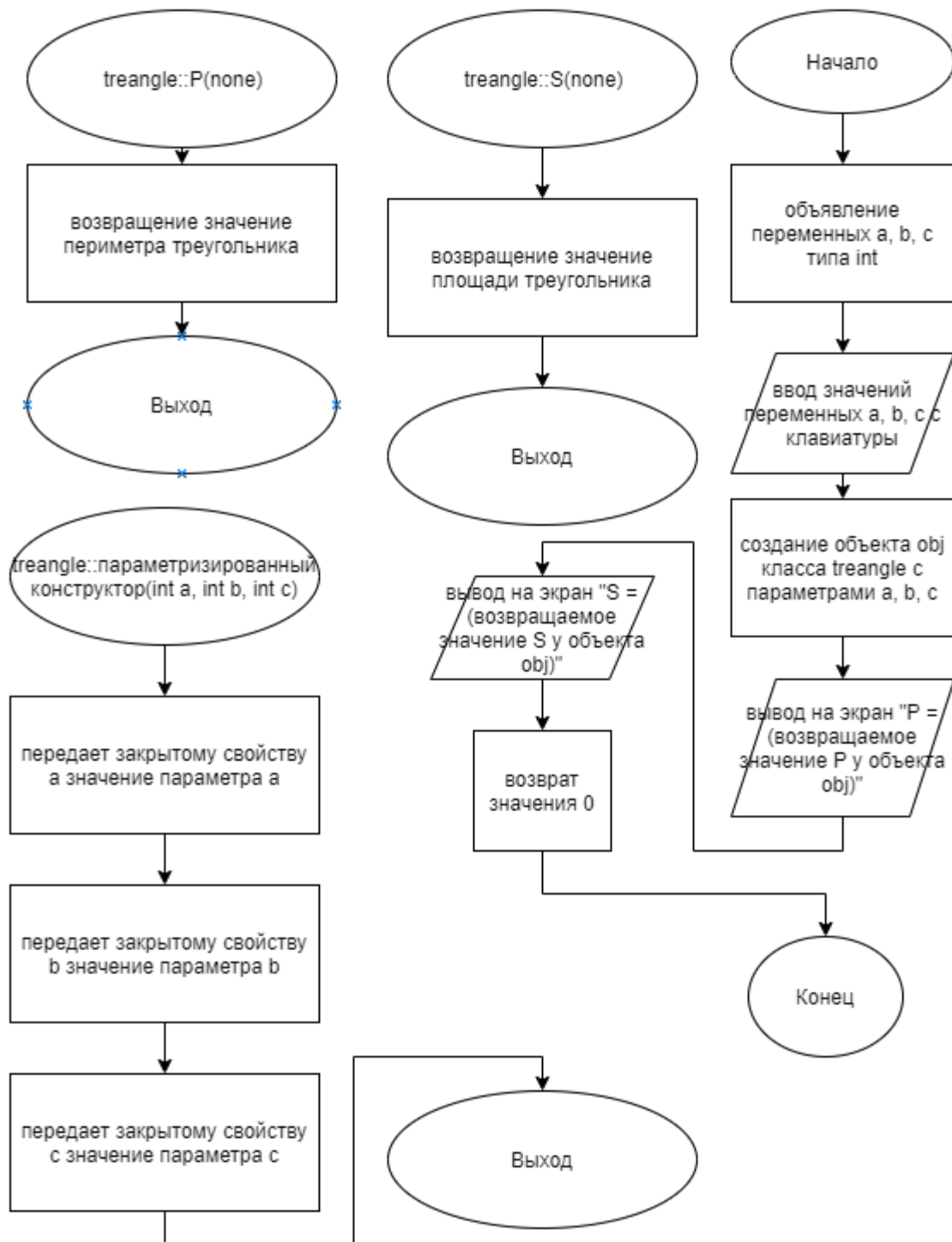


Рисунок 1 – Блок-схема алгоритма

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 "treangle.h"

using namespace std;

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

5.2 Файл treangle.cpp

Листинг 2 – treangle.cpp

```
#include "treangle.h"
#include <iostream>
#include <math.h>

using namespace std;

treangle::treangle(int a, int b, int c)
{
    this->a=a;
    this->b=b;
    this->c=c;
}
int treangle::P()
```

```

    {
        return a+b+c;
    }
double triangle::S()
{
    double p = (a+b+c)/2.0;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}

```

5.3 Файл `triangle.h`

Листинг 3 – `triangle.h`

```

#ifndef __TREANGLE__H
#define __TREANGLE__H

class triangle
{
    private:
        int a;
        int b;
        int c;
    public:
        triangle(int a, int b, int c);
        int 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
1 1 1	P = 3 S = 0.433013	P = 3 S = 0.433013

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).