

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 calculatePerimeter класса Triangle.....	9
3.3 Алгоритм метода calculateArea класса Triangle.....	10
3.4 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл Triangle.cpp.....	14
5.3 Файл Triangle.h.....	15
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17

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 хранит длины сторон треугольника и обеспечивает методы для вычисления периметра и площади треугольника;
- объект `cout` класса потокового вывода предназначен для предназначен для функционирования системы;
- объект `cin` класса потокового ввода предназначен для предназначен для функционирования системы;
- функция `main` для основной алгоритм работы программы.

Класс `Triangle`:

- свойства/поля:
 - поле первая сторона треугольника:
 - наименование — `side1`;
 - тип — целочисленный;
 - модификатор доступа — `private`;
 - поле вторая сторона треугольника:
 - наименование — `side2`;
 - тип — целочисленный;
 - модификатор доступа — `private`;
 - поле третья сторона треугольника:
 - наименование — `side3`;
 - тип — целочисленный;
 - модификатор доступа — `private`;
- функционал:
 - метод `Triangle` — конструктор;

- o метод `calculatePerimeter` — вычисление периметра треугольника;
- o метод `calculateArea` — 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: конструктор.

Параметры: s1, s2, s3 - длины сторон треугольника.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Triangle

№	Предикат	Действия	№ перехода
1		определяется конструктор класса "Triangle", который принимает три целых значения в качестве параметров	2
2		присваивается значение параметра 's1' переменной 'side1' внутри конструктора класса 'Triangle'	3
3		присваивается значение параметра 's2' переменной 'side2' внутри конструктора класса 'Triangle'	4
4		присваивается значение параметра 's3' переменной 'side3' внутри конструктора класса 'Triangle'	Ø

3.2 Алгоритм метода calculatePerimeter класса Triangle

Функционал: вычисление периметра треугольника.

Параметры: отсутствуют.

Возвращаемое значение: целочисленное значение, равное значению периметру треугольника.

Алгоритм метода представлен в таблице 2.

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

№	Предикат	Действия	№ перехода
1		определение метода 'calculatePerimeter()' класса 'Triangle'	2
2		вычисление суммы длин сторон треугольника и ее возврат	∅

3.3 Алгоритм метода *calculateArea* класса *Triangle*

Функционал: вычисление площади треугольника.

Параметры: отсутствуют.

Возвращаемое значение: значение типа 'double' - площадь треугольника.

Алгоритм метода представлен в таблице 3.

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

№	Предикат	Действия	№ перехода
1		определение метода 'calculateArea()' класса 'Triangle'	2
2		инициализация переменной 's' значением полупериметра треугольника, которое вычисляется как половина значения, возвращаемого методом 'calculatePerimeter()'	3
3		вычисление площади треугольника используя формулу Герона	∅

3.4 Алгоритм функции *main*

Функционал: основной алгоритм работы программы.

Параметры: отсутствуют.

Возвращаемое значение: целочисленный индикатор корректности завершения работы.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции *main*

№	Предикат	Действия	№
---	----------	----------	---

			перехода
1		объявление целочисленных переменных s1, s2, s3	2
2		ввод значения переменных s1, s2, s3 с клавиатуры	3
3		создается объект класса 'Triangle' с именем 'triangle', используя конструктор с передачей аргументов 's1', 's2', 's3'	4
4		вывод на экран "P = ", значение периметра треугольника	5
5		вывод на экран "S = ", значение площади треугольника	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

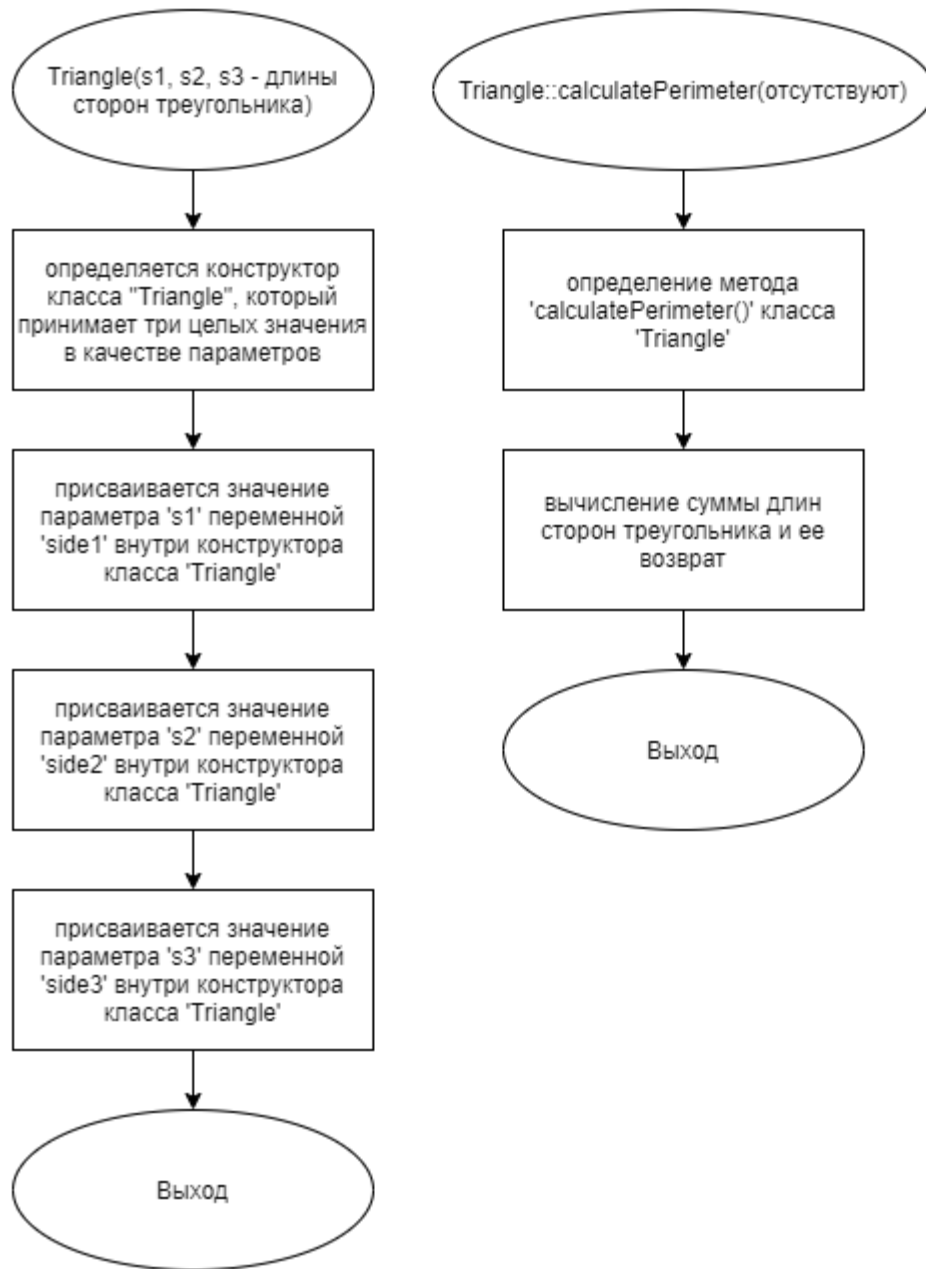


Рисунок 1 – Блок-схема алгоритма

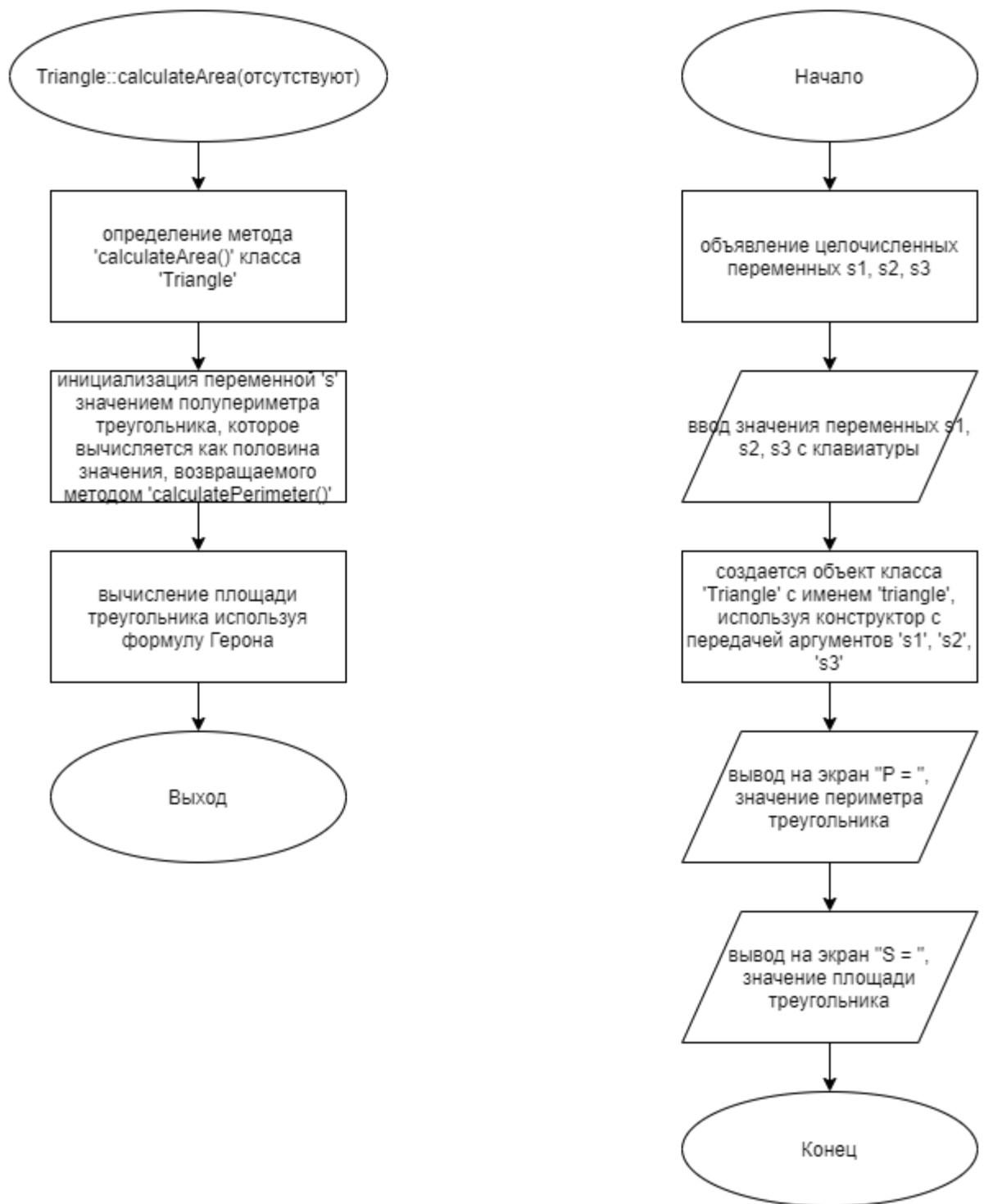


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <iostream>
#include <stdlib.h>
#include <stdio.h>
#include "Triangle.h"

using namespace std;

int main() {
    int s1, s2, s3;

    cin >> s1 >> s2 >> s3;

    Triangle triangle(s1, s2, s3);

    cout << "P = " << triangle.calculatePerimeter() << endl;
    cout << "S = " << triangle.calculateArea() << endl;

    return 0;
}
```

5.2 Файл Triangle.cpp

Листинг 2 – Triangle.cpp

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

Triangle::Triangle(int s1, int s2, int s3) {
    side1 = s1;
    side2 = s2;
    side3 = s3;
}

int Triangle::calculatePerimeter() {
```

```
        return side1 + side2 + side3;
    }

    double Triangle::calculateArea() {
        double s = calculatePerimeter() / 2.0;
        return sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }
```

5.3 Файл Triangle.h

Листинг 3 – Triangle.h

```
#ifndef __TRIANGLE__H
#define __TRIANGLE__H

class Triangle{
private:
    int side1, side2, side3;

public:
    Triangle(int s1, int s2, int s3);
    int calculatePerimeter();
    double calculateArea();
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 10 8	P = 23 S = 19.81	P = 23 S = 19.81

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).