

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 конструктора класса Triangle.....	8
3.2 Алгоритм метода perimetr класса Triangle.....	8
3.3 Алгоритм функции main.....	8
3.4 Алгоритм метода square класса Triangle.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	10
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 ABC класса Triangle;
- функция main для основная функция;
- Объект стандартного ввода cin;
- Объект стандартного вывода cout.

Класс Triangle:

- свойства/поля:
 - поле Сторона:
 - наименование — a;
 - тип — целое;
 - модификатор доступа — private;
 - поле Строна:
 - наименование — b;
 - тип — целое;
 - модификатор доступа — private;
 - поле строны:
 - наименование — c;
 - тип — целое;
 - модификатор доступа — private;
- функционал:
 - метод Triangle — конструктор;
 - метод perimetr — ;
 - метод square — .

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: Параметрический конструктор.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса *Triangle*

№	Предикат	Действия	№ перехода
1		Создание объекта	Ø

3.2 Алгоритм метода **perimetr** класса **Triangle**

Функционал: Вычисление периметра.

Параметры: нет.

Возвращаемое значение: целое, периметр.

Алгоритм метода представлен в таблице 2.

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

№	Предикат	Действия	№ перехода
1		Создание объекта	Ø

3.3 Алгоритм функции **main**

Функционал: основная функция.

Параметры: нет.

Возвращаемое значение: Целое, код успеха.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление трех целочисленных переменных a, b, c	2
2		Ввод переменных a, b, c	3
3		Объявление переменной obj типа Triangle от переменных a, b, c	4
4		Ввод периметра и площади	Ø

3.4 Алгоритм метода *square* класса *Triangle*

Функционал: Вычисление площади.

Параметры: нет.

Возвращаемое значение: Действительное, площадь.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *square* класса *Triangle*

№	Предикат	Действия	№ перехода
1		Созданиеи объекта	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

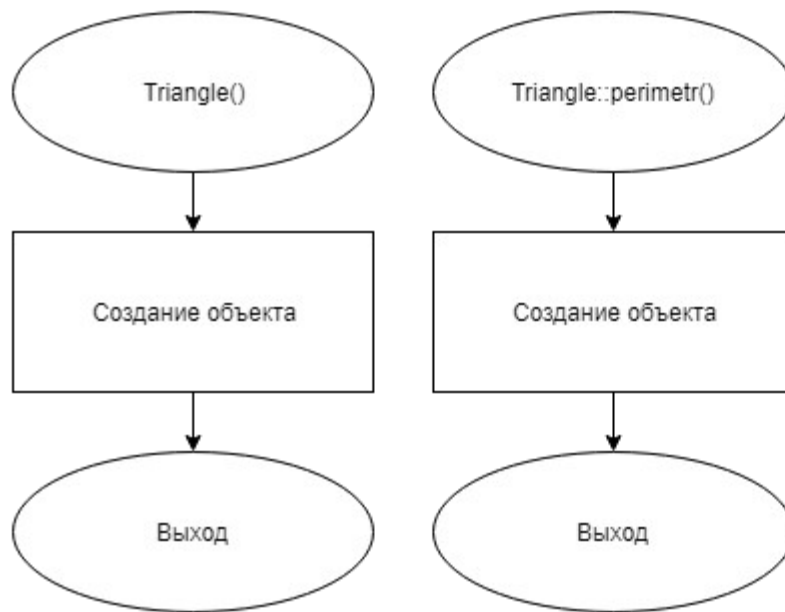


Рисунок 1 – Блок-схема алгоритма



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

using namespace std;

int main()
{
    int a, b, c;
    cin >> a >> b >> c;
    Triangle ABC(a, b, c);
    cout << "P = " << ABC.perimetr() << endl;
    cout << "S = " << ABC.square();

    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;
}
long long int Triangle::perimetr()
{
    return (a + c + b);
}
```

```
long double Triangle::square()
{
    long 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 __TRIANGLE__H
#define __TRIANGLE__H
#include <iostream>
using namespace std;
class Triangle
{
    int a, b, c;
public:
    Triangle(int, int, int);
    long long int perimetr();
    long double square();
};

#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).