

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 функции main.....	8
3.2 Алгоритм конструктора класса Triangle.....	8
3.3 Алгоритм деструктора класса Triangle.....	9
3.4 Алгоритм метода Per класса Triangle.....	9
3.5 Алгоритм метода Square класса Triangle.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
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 tr класса Triangle;
- cin - объект стандартного потока ввода;
- cout - объект стандартного потока вывода.

Класс Triangle:

- свойства/поля:
 - поле сторона a:
 - наименование — a;
 - тип — int;
 - модификатор доступа — private;
 - поле сторона b:
 - наименование — b;
 - тип — int;
 - модификатор доступа — private;
 - поле сторона c:
 - наименование — c;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод Triangle — Конструктор;
 - метод ~Triangle — Деструктор;
 - метод Per — Вычисление периметра;
 - метод Square — Вычисление площади.

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 функции `main`

Функционал: основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: целочисленное - индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции `main`

№	Предикат	Действия	№ перехода
1		объявление целочисленных переменных <code>a1</code> , <code>b1</code> , <code>c1</code> .	2
2		Ввод значений переменных <code>a1</code> , <code>b1</code> , <code>c1</code> с клавиатуры	3
3		Создание объекта <code>tr</code> класса <code>Triangle</code> с передачей параметров <code>a1</code> , <code>b1</code> , <code>c1</code>	4
4		Вывод с первой строки периметра, со второй строки площади треугольника с помощью методов <code>Per</code> и <code>Square</code> класса <code>Triangle</code> соответственно	Ø

3.2 Алгоритм конструктора класса `Triangle`

Функционал: Конструктор.

Параметры: `int a1` - сторона `a`, `int b1` - сторона `b`, `int c1` - сторона `c`.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса *Triangle*

№	Предикат	Действия	№ перехода
1		присвоение а значения a1	2
2		присвоение b значения b1	3
3		присвоение с значения c1	∅

3.3 Алгоритм деструктора класса *Triangle*

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса *Triangle*

№	Предикат	Действия	№ перехода
1		Уничтожение объекта	∅

3.4 Алгоритм метода *Per* класса *Triangle*

Функционал: Вычисление периметра.

Параметры: нет.

Возвращаемое значение: Double - периметр.

Алгоритм метода представлен в таблице 4.

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

№	Предикат	Действия	№ перехода
1		Инициализация периметра P, равного сумме полей	2
2		Возврат значения P	∅

3.5 Алгоритм метода Square класса Triangle

Функционал: Вычисление площади.

Параметры: нет.

Возвращаемое значение: Double - площадь.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Square класса Triangle

№	Предикат	Действия	№ перехода
1		Инициализация переменной P1 - полупериметр	2
2		Инициализация переменной S - площадь, которая равна $\sqrt{P1*(P1-a)*(P1-b)*(P1-c)}$	3
3		Возврат значения S	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

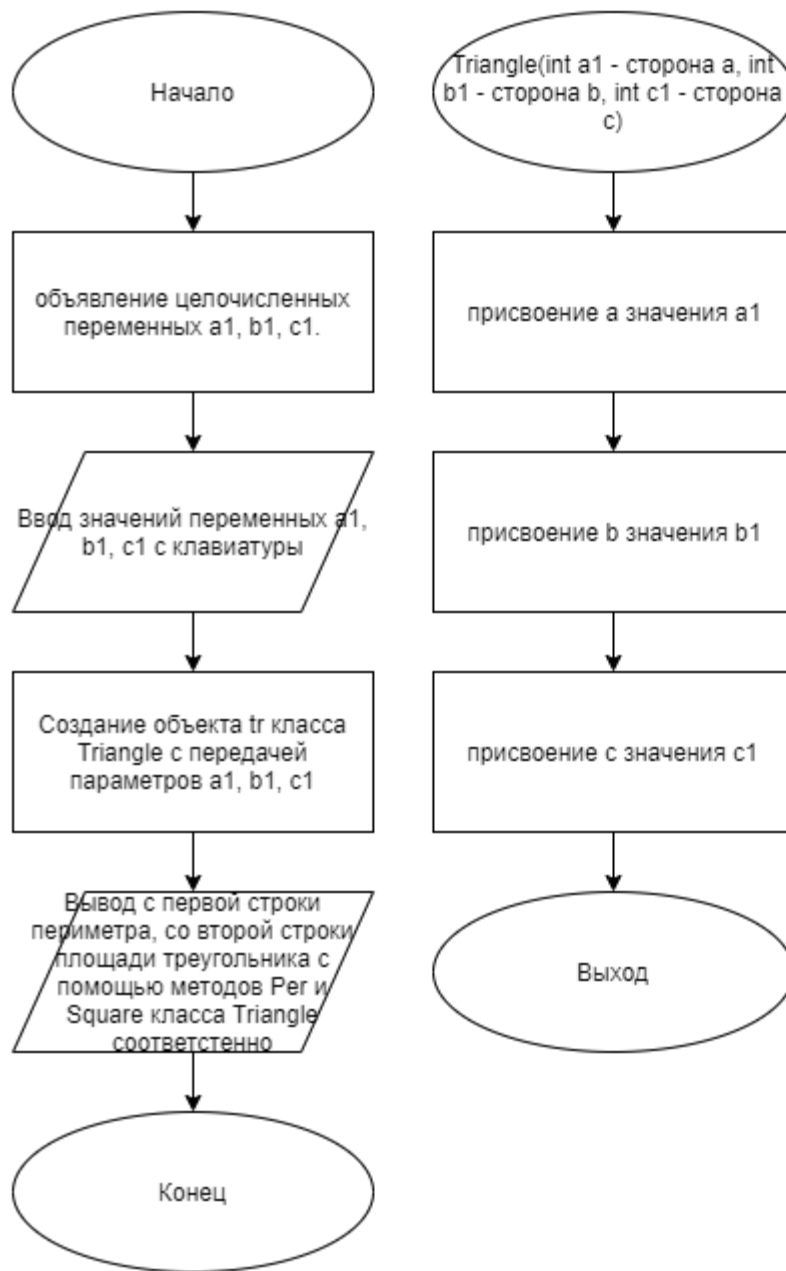


Рисунок 1 – Блок-схема алгоритма

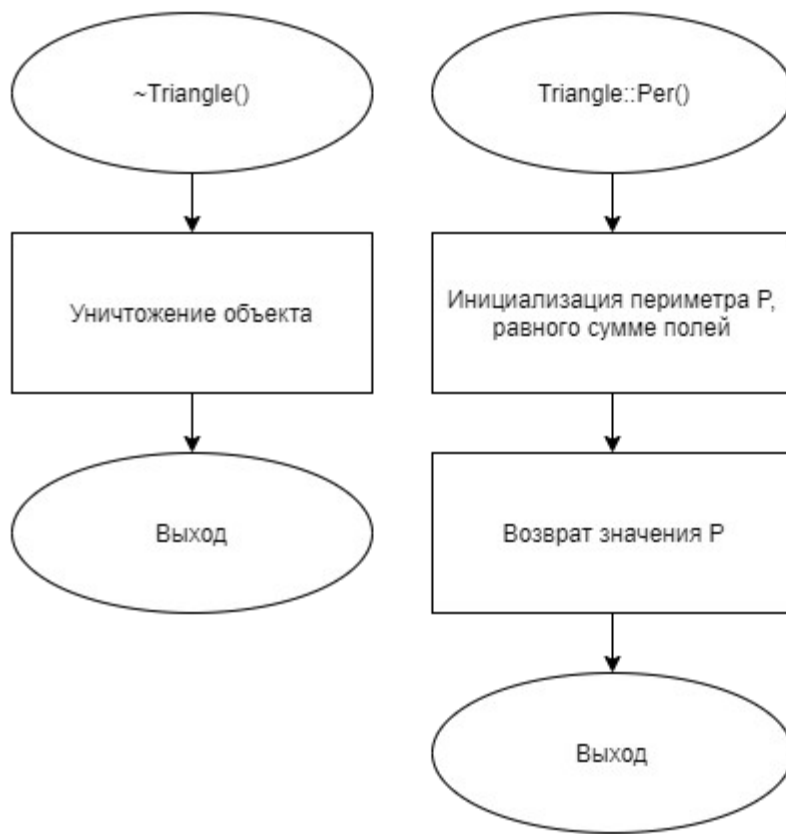


Рисунок 2 – Блок-схема алгоритма



Рисунок 3 – 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 a1,b1,c1;
    cin >> a1;
    cin >> b1;
    cin >> c1;
    Triangle tr(a1,b1,c1);
    cout<<"P = "<<tr.Per()<<endl<<"S = "<<tr.Square(tr.Per());
    return(0);
}
```

5.2 Файл Triangle.cpp

Листинг 2 – Triangle.cpp

```
#include "Triangle.h"
#include <iostream>
#include <math.h>
Triangle::Triangle(int a1, int b1, int c1){
    a=a1;
    b=b1;
    c=c1;
}
double Triangle::Per(){
    double P = a+b+c;
    return P;
}
double Triangle::Square(double P){
    double P1=P/2;
```

```
        double S = sqrt(P1*(P1-a)*(P1-b)*(P1-c));  
        return S;  
    }  
    Triangle::~~Triangle(){}  
}
```

5.3 Файл Triangle.h

Листинг 3 – Triangle.h

```
#ifndef __TRIANGLE__H  
#define __TRIANGLE__H  
class Triangle  
{  
    private:  
        int a;  
        int b;  
        int c;  
    public:  
        Triangle(int a1, int b1, int c1);  
        double Per( ) ;  
        double Square(double P) ;  
        ~Triangle();  
};  
#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
4 5 6	P = 15 S = 9.92157	P = 15 S = 9.92157
5 6 7	P = 18 S = 14.6969	P = 18 S = 14.6969

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).