ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА» (РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 2

По дисциплине «Языки программирования»

Выполнил: ст. гр. ТКИ – 141

Дьячков Денис Андреевич

Проверил: к.т.н., доц.

Васильева М. А.

Москва 2022.

Содержание.

1. Формулировка задачи

2. Блок-схема алгоритма

3. Решение задачи на языке программирования C++

4. Решение тестовых примеров

5. Решение тестовых примеров в Excel

6.Зачет задания в Github

Задание 2-1

**1.Формулировка задачи**

|  |  |
| --- | --- |
| **Вариант** | **Задача** |
| №2 | По заданным длинам двух катетов прямоугольного треугольника вычислить его:  1)периметр  2)площадь |

**2.Блок-схема алгоритма**

Блок-схемы алгоритмов функций представлены на рисунках (Рисунок 1, Рисунок 2,).

perimetr=getPerimetr(first,second)

area=getArea(first,second)

first,second

окончание

**Рисунок 1-Блок-схема функции main()**

first,second

Perimetr=

Perimetr

**Рисунок 2-Блок схема функции getPerimetr()**

first,second

Area=(first\*second)/2

Area

**Рисунок 3-Блок схема функции getArea()**

**3.Решение задачи на языке программирования C++**

#defline USE\_MATH\_DEFINES

#include <iostream>

#include <cmath>

using namespace std;

/\*\*

\* \brief Рассчитывает периметр треугольника.

\* \param first первый катет прямоугольного треугольника.

\* \param second второй катет прямоугольного треугольника.

\* \return периметр треугольника.

\*/

double getPerimetr(double first, double second);

/\*\*

\* \brief Рассчитывает площадь треугольника.

\* \param first первый катет прямоугольного треугольника.

\* \param second второй катет прямоугольного треугольника.

\* \return площадь треугольника.

\*/

double getArea(double first, double second);

/\*\*

\* \brief считать сторону трегольника из консоли.

\* \param message побуждающая надпись для пользователя.

\* \return сторона треугольника треугольника.

\*/

double getSide( const string& message = "");

/\*\*

\* \brief пользовательский ввод

\* \param PERIMETR расчет периметра треугольника

\* \param AREA расчет площади треугольника

\*/

enum class userInput

{

PERIMETR = 1,

AREA

};

/\*\*

\* \brief

\* \return

\*/

int main()

{

try

{

const double first = getSide("Введите значение первого катета треугольника = ");

const double second = getSide("Введите значение второго катета треугольника = ");

cout « "Введите число,соответствующее заданию.\n"

« static\_cast<int>(userInput::AREA) « "- вычесление площади треугольника.\n "

« static\_cast<int>(userInput::PERIMETR) « "- вычесление периметра треугольника.\n "

int input = 0;

cin » input;

const auto choice = static\_cast<userInput>(input);

switch(choice)

{

case userInput::AREA:

{

const auto area = getArea(first,second);

cout « "Площадь треугольника = " « area;

break;

}

case userInput::PERIMETR:

{

const auto perimetr = getPerimetr(first,second);

cout « "Периметр треугольника = " « perimetr;

break;

}

}

}

catch (out\_of\_range&)

{

cout « "Введена неправильная сторона!";

return 1;

}

return 0;

}

double getPerimetr(const double first, const double second)

{

return first + second + sqrt(pow(first,2) + pow(second,2))

}

double getArea(const double first, const double second)

{

return (first \* second)/2

}

double getSide(const string& message)

{

cout « message;

double side = 0.0;

cin » side;

if (side < = 0)

{

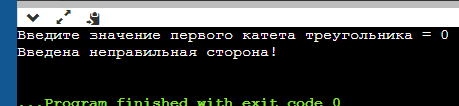
throw out\_of\_range("Введена неправильная сторона!");

}

return side;

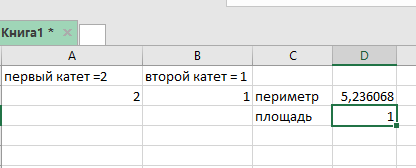
}

**4. Решение тестовых примеров**

****

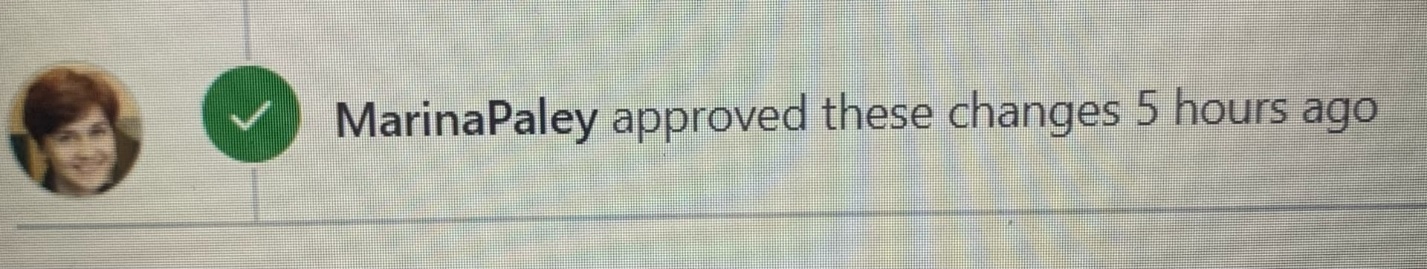
**Рисунок 4 – Решение тестового примера(пример работы при неправильной величине)**

**5. Решение тестовых примеров в Excel**

****

**Рисунок 5- Решение в Excel**

**6. Зачет задания в GitHub**

****

**Рисунок 6 – Зачет задания**