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 ЛАБОРАТОРНОЙ РАБОТЕ №1

По дисциплине «Языки программирования»

ВАРИАНТ 10

Выполнил: ст. гр. ТКИ - 142

Левин Всеволод Максимович

Проверил: к.т.н., доц. Васильева М. А.

(Проверил: к.т.н, доц. Балакина Е. П.)

Москва 2023

Оглавление

[**1 Решение задачи 1.1 3**](#_heading=h.gjdgxs)

[1.1 Формулировка задачи 3](#_heading=h.30j0zll)

[1.2 Блок-схема алгоритма 4](#_heading=h.1fob9te)

[1.3 Код для задания 1.1 6](#_heading=h.3dy6vkm)

[1.4 Решение различных тестовых примеров на С 7](#_heading=h.1t3h5sf)

[1.5 Решение различных тестовых примеров в сторонних приложениях 7](#_heading=h.2s8eyo1)

[1.6 Подтверждение approve 1.1 8](#_heading=h.3rdcrjn)

[**2 Решение задачи 1.2 9**](#_heading=h.lnxbz9)

[2.1 Формулировка задачи 9](#_heading=h.35nkun2)

[2.2 Блок-схема алгоритма 9](#_heading=h.1ksv4uv)

[2.3 Код для задачи 1.2 13](#_heading=h.44sinio)

[2.4 Поясняющие формулы 17](#_heading=h.2jxsxqh)

[2.5 Решение различных тестовых примеров на С 17](#_heading=h.z337ya)

[2.6 Решение различных тестовых примеров в сторонних приложениях 18](#_heading=h.3j2qqm3)

[2.7 Подтверждение approve 1.2 19](#_heading=h.1y810tw)

1. **Решение задачи 1.1**
   1. **Формулировка задачи**

Создать консольное приложение, вычисляющее значения переменных по представленным в таблице формулам. Расчёт примера осуществить по заданным константам. Вывести на экран значения исходных данных, а также результат вычислений. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 1 – Условие задачи 1.1

| **Вариант** | **Формулы** | **Константы** |
| --- | --- | --- |
| **10** |  | x=-4.8  y=17.5  z=3.2 |

* 1. **Блок-схема алгоритма**

Блок-схема алгоритма представлена ниже на Рис. 1, Рис. 2, Рис. 3

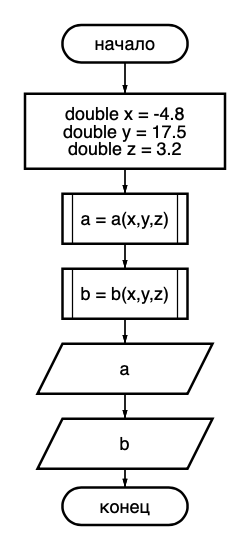


Рис. 1. Блок-схема функции main()

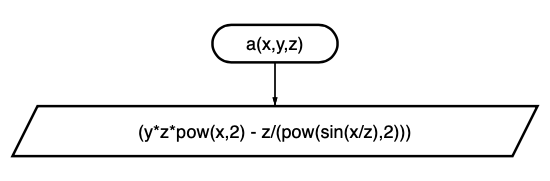


Рис. 2. Блок-схема функции a(x,y,z)

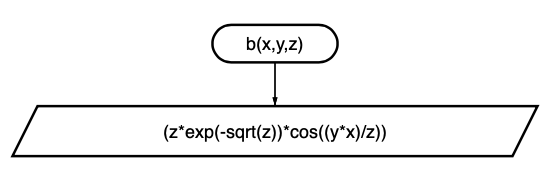


Рис. 3. Блок-схема функции b(x,y,z)

* 1. **Код для задания 1.1**

#include <stdio.h>

#include <math.h>

/\*\*

\* @breaf расчитывает значение переменной a по формуле

\* @param x - значение параметра x

\* @param y - значение параметра y

\* @param z - значение параметра z

\* @return возвращает значение переменной a по формуле

\*/

double a(double x, double y, double z);

/\*\*

\* @breaf рассчитывает значение переменной b по формуле

\* @param x - значение параметра x

\* @param y - значение параметра y

\* @param z - значение параметра z

\* @return возвращает значение переменной b по формуле

\*/

double b(double x, double y, double z);

/\*\*

\* @breaf Точка входа в программу

\* @return Возвращает 0, если ошибок нет, не 0, если ошибки присутствуют.

\*/

int main()

{

const double x = -4.8;

const double y = 17.5;

const double z = 3.2;

printf("result of a = %f", a(x,y,z));

printf("\nresult of b = %f", b(x,y,z));

return 0;

}

double a(double x, double y, double z)

{

return (y\*z\*pow(x, 2) - (z/(pow(sin(x/z),2))));

}

double b(double x, double y, double z)

{

return (z\*exp(-sqrt(z))\*cos((y\*x)/z));

}

* 1. **Решение различных тестовых примеров на С**

Ниже представлен вывод программы 1.1 в С (Рис. 4)

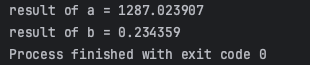


Рис. 4. Вывод программы 1.1

* 1. **Решение различных тестовых примеров в сторонних приложениях**

Ниже представлено решение задания 1.1 в приложении “Photomath” (Рис. 5)

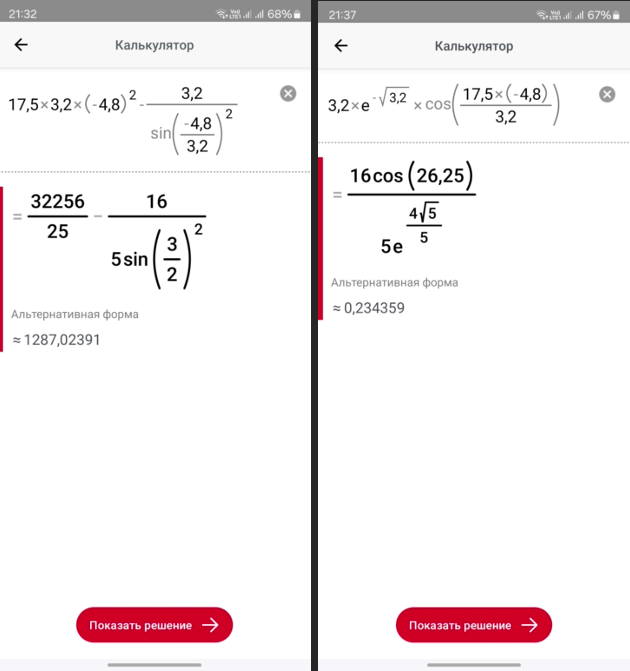


Рис. 5. Решение задания 1.1 в приложении “Photomath”

* 1. **Подтверждение approve 1.1**

На рис. 6 представлено подтверждение (approve) задания 1.1 преподавателем.

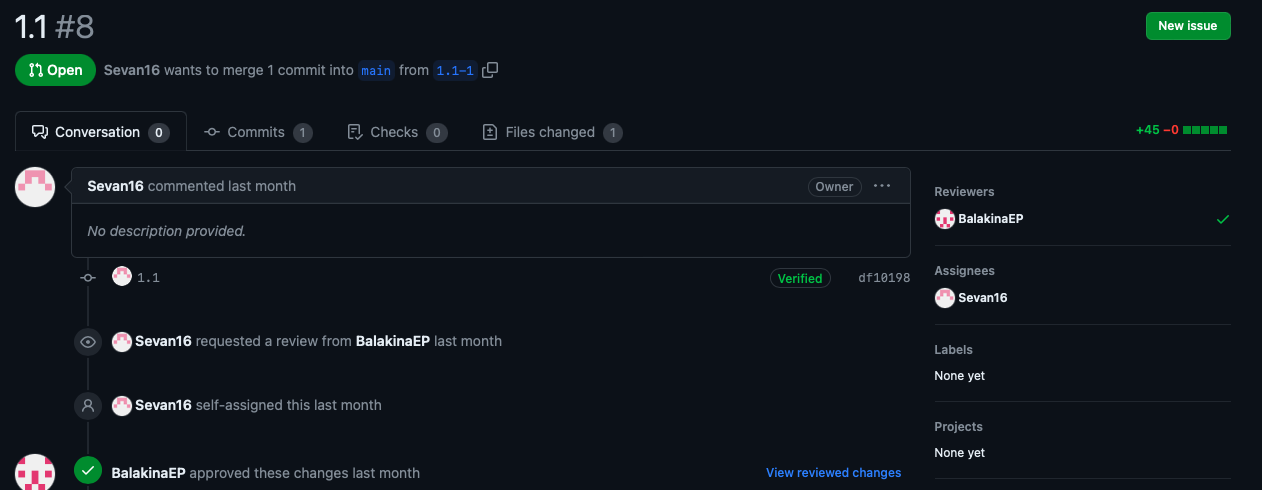


Рис. 6. Approve задачи 1.1

1. **Решение задачи 1.2**
   1. **Формулировка задачи**

Создать консольное приложение для решения задачи, представленной в таблице. Данные для решения вводит пользователь. Помните, что ввод необходимо проверять на правильность (только числа). Вывести результат вычислений на экран. При необходимости дополнить свой отчёт поясняющими формулами, помогающими решить задачу. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 2 – Условие задачи 1.2

| Вариант | Задача |
| --- | --- |
| 10 | Вычислить площади геометрических фигур: прямоугольника и треугольника по заданным сторонам. |

* 1. **Блок-схема алгоритма**

Блок-схемы алгоритма функции main, а также внешних функций представлены ниже на рисунках.

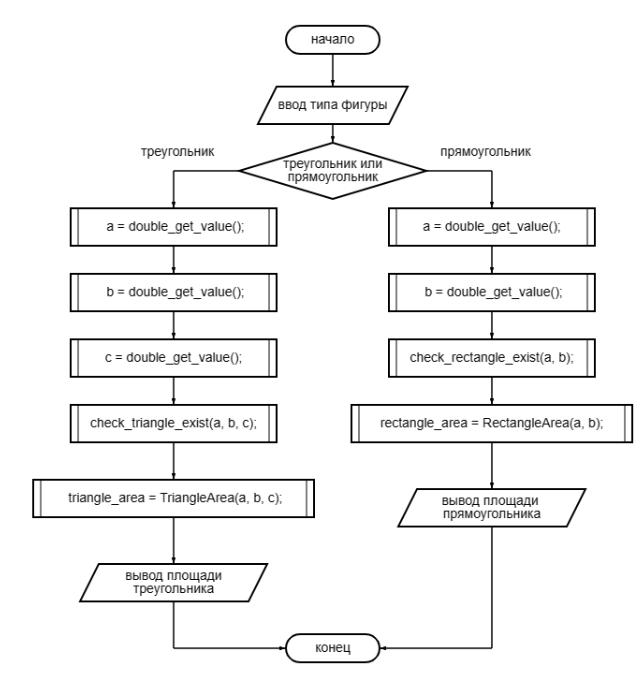


Рис. 7. Блок-схема функции main()

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рис. 8. Блок-схема функции check\_triangle\_exist(a,b,c)

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рис. 9. Блок-схема функции check\_rectangle\_exist(a, b)

Изображение выглядит как текст, Шрифт, линия, диаграмма

Автоматически созданное описание

Рис. 10. Блок-схема функции RectangleArea(a,b)

Изображение выглядит как текст, Шрифт, диаграмма, линия

Автоматически созданное описание

Рис. 11. Блок-схема функции TriangeArea(a,b)

Изображение выглядит как диаграмма, текст, зарисовка, линия

Автоматически созданное описание

Рис. 12. Блок-схема функци double\_get\_value()

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рис. 13. Блок-схема функции int\_get\_value()

* 1. **Код для задачи 1.2**

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#include <errno.h>

/\*\*

\* @breaf проверяет правильность ввода данных

\* @return возвращает результат в случае успеха

\*/

double double\_get\_value();

/\*\*

\* @breaf вычисляет площадь треугольника

\* @param a – первая сторона треугольнка

\* @param b – вторая сторона треугольника

\* @param c – третья сторона треугольника

\* @return возвращает площадь треугольника

\*/

double TriangleArea(double a, double b, double c);

/\*\*

\* @breaf вычисляет площадь прямоугольника

\* @param a – первая сторона прямоугольника

\* @param b – вторая сторона прямоугольника

\* @return возвращает площадь прямоугольника

\*/

double RectangleArea(double a, double b);

/\*\*

\* @breaf проверяет правильность ввода данных

\* @return возвращает результат в случае успеха

\*/

int int\_get\_value();

/\*\*

\* @breaf Проверяет существование треугольника по заданным сторонам

\* @param a – первая сторона треугольника

\* @param b – вторая сторона треугольника

\* @param c – третья сторона треугольника

\*/

void check\_triangle\_exist(double a, double b, double c);

/\*\*

\* @breaf проверяет существование прямоугольника по заданным сторонам

\* @param a – первая сторона прямоугольника

\* @param b – вторая сторона прямоугольника

\*/

void check\_rectangle\_exist(double a, double b);

/\*\*

\* @brief Выбор типа фигуры.

\*/

enum shape\_type {

/\*\*

\* @brief Выбор треугольника.

\*/

Triangle = 1,

/\*\*

\* @brief Выбор прямоугольника.

\*/

Rectangle

};

int main()

{

printf("Выберите тип фигуры:\n");

printf("1. Треугольник\n");

printf("2. Прямоугольник\n");

int choose = int\_get\_value();

enum shape\_type user\_input = (enum shape\_type)choose;

switch (user\_input)

{

case Triangle:

printf("Введите первую сторону треугольника: ");

double a = double\_get\_value();

printf("Введите вторую сторону треугольника: ");

double b = double\_get\_value();

printf("Введите третью сторону треугольника: ");

double c = double\_get\_value();

check\_triangle\_exist(a, b, c);

double triangle\_area = TriangleArea(a, b, c);

printf("Площадь заданного треугольника равна: %lf", triangle\_area);

break;

case Rectangle:

printf("Введите первую сторону прямоугольника: ");

a = double\_get\_value();

printf("Введите вторую сторону прямоугольника: ");

b = double\_get\_value();

check\_rectangle\_exist(a, b);

double rectangle\_area = RectangleArea(a, b);

printf("Площадь заданного прямоугольника равна: %lf", rectangle\_area);

break;

}

return 0;

}

double TriangleArea(double a, double b, double c)

{

double p = (a + b + c) / 2.0;

return sqrt(p \* (p - a) \* (p - b) \* (p - c));

}

double RectangleArea(double a, double b)

{

return a \* b;

}

double double\_get\_value()

{

double result;

double scanf\_result = scanf("%lf", &result);

if (scanf\_result!=1)

{

errno = EIO;

perror("Input/Output error.");

abort();

}

return result;

}

int int\_get\_value()

{

int result;

int scanf\_result = scanf("%d", &result);

if (scanf\_result!=1)

{

errno = EIO;

perror("Input/Output error.");

abort();

}

return result;

}

void check\_triangle\_exist(double a, double b, double c)

{

if ((a <= 0) || (b <= 0) || (c <= 0))

{

errno = EIO;

perror("Введите положительные стороны треугольника.");

abort();

}

if ((a > b + c) || (b > a + c) || (c > b + a))

{

errno = EIO;

perror("Треугольника с заданными сторонами не существует.");

abort();

}

}

void check\_rectangle\_exist(double a, double b)

{

if (a <= 0 || b <= 0)

{

errno = EIO;

perror("Прямоугольника с заданными сторонами не существует.");

abort();

}

}

* 1. **Поясняющие формулы**

На рис. 14 представлена формула площади треугольника (формула Герона), где S – площадь, p – полупериметр, a,b,c – стороны треугольника.



Рис. 14. Формула Герона

На рис. 15 представлена формула площади прямоугольника, где S – площадь, a,b – стороны прямоугольника.

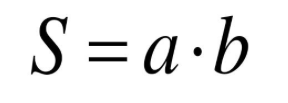


Рис. 15. Формула площади прямоугольника

* 1. **Решение различных тестовых примеров на С**

На рис. 16 представлен тест программы (выбран треугольник со сторонами 3, 4, 5)

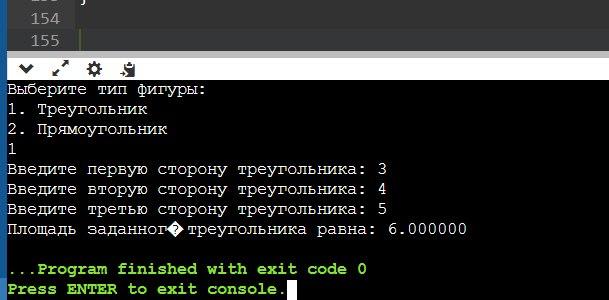


Рис. 16. Тест программы (треугольник)

На рис. 17 представлен тест программы (выбран прямоугольник со сторонами 4, 5)

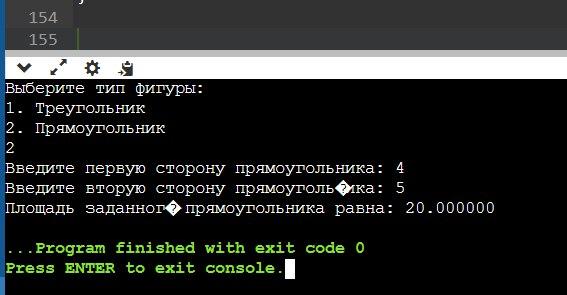


Рис. 17. Тест программы (прямоугольник)

* 1. **Решение различных тестовых примеров в сторонних приложениях**

На рис. 18 представлено вычисление площади треугольника по формуле Герона с помощью онлайн калькулятора.

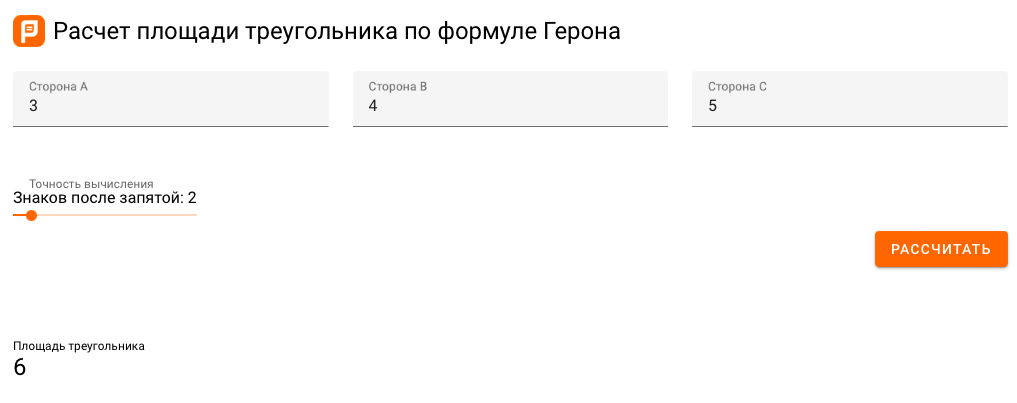


Рис. 18. Вычисление площади треугольника по формуле Герона с помощью онлайн калькулятора

На рис. 19 представлено вычисление площади прямоугольника с помощью онлайн калькулятора.

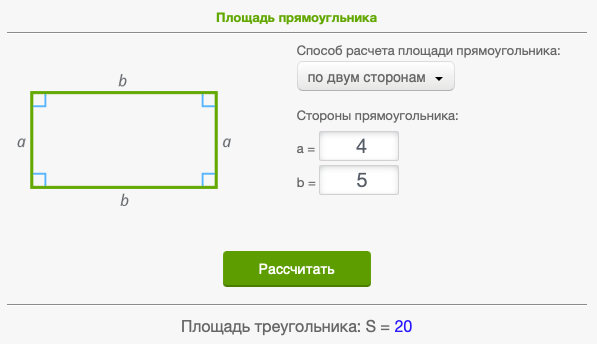


Рис. 19. Вычисление площади прямоугольника с помощью онлайн калькулятора

* 1. **Подтверждение approve 1.2**

На рис. 20 представлено подтверждение (approve) задания 1.2 преподавателем.

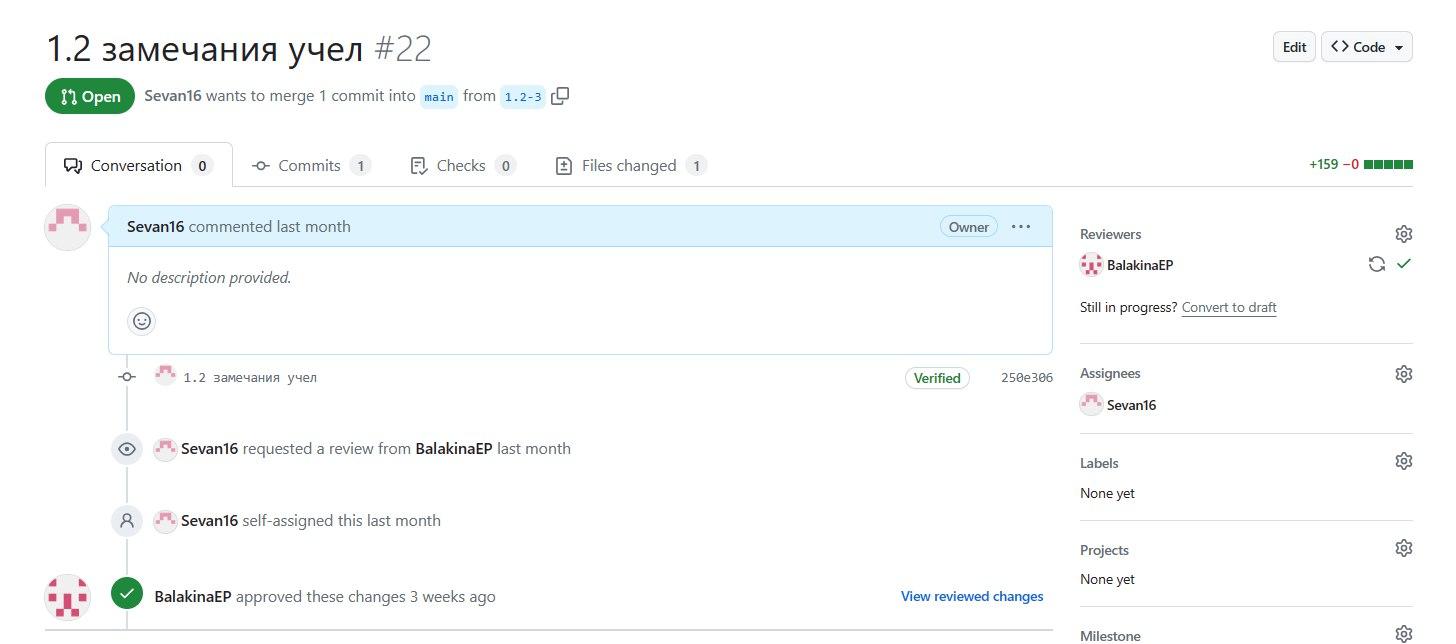


Рис. 20. Подтверждение (approve) задания 1.2 преподавателем.