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 6

Выполнил: ст. гр. ТКИ-141

Винтфельд Рина Дмитриевна

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

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

Москва 2023

1. Задача 1.1
   1. Формулировка задания 1.1

Создать консольное приложение, вычисляющее значения переменных по представленным в таблице формулам (Таблица 1). Расчёт примера осуществить по заданным константам. Вывести на экран значения исходных данных, а также результат вычислений. Дополнить свой отчёт блок-схемой алгоритма.

Таблица  – Исходные данные

|  |  |  |
| --- | --- | --- |
| Вариант | Формулы | Константы |
| 6 |  | x=0.61  y=3.4  z=16.5 |

* 1. Блок-схема алгоритма 1.1

Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы функций расчета значений a и b представлены ниже (Рисунок 2).

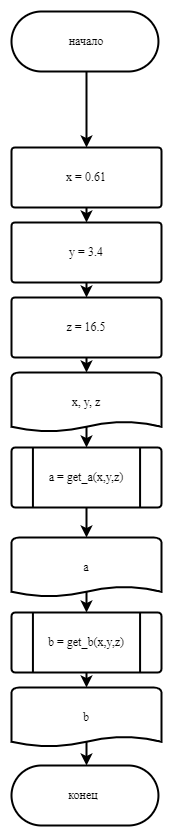


Рисунок  ­- Блок-схема основного алгоритма

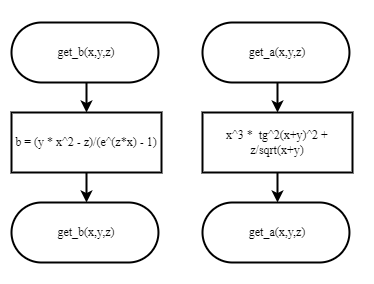


Рисунок  – Блок-схема используемых функций

* 1. Текст программы на языке C 1.1

#include <stdio.h>

#include <math.h>

/\*\*

\*@brief формула для получения переменной а

\*@return возвращает значение a

\*/

double get\_a(double const x, double const y, double const z);

/\*\*

\*@brief формула для получения переменной b

\*@return возвращает значение b

\*/

double get\_b(double const x, double const y, double const z);

/\*\*

\*@brief точка входа в программу

\*@return 0 в случае работы программы

\*/

int main()

{

double const x = 0.61;

double const y = 3.4;

double const z = 16.5;

printf\_s("x = %lf\ny = %lf\nz = %lf", x, y, z);

printf\_s("\na = %lf", get\_a(x, y, z));

printf\_s("\nb = %lf", get\_b(x, y, z));

return 0;

}

double get\_a(double const x, double const y, double const z)

{

return pow(x, 3) \* pow(tan(pow(x + y, 2)), 2) + z / sqrt(x + y);

}

double get\_b(double const x, double const y, double const z)

{

return (y \* pow(x, 2) - z) / (exp(x \* z) - 1);

}

* 1. Результаты выполнения программы 1.1

Результаты выполнения программы представлен ниже (Рисунок 3).

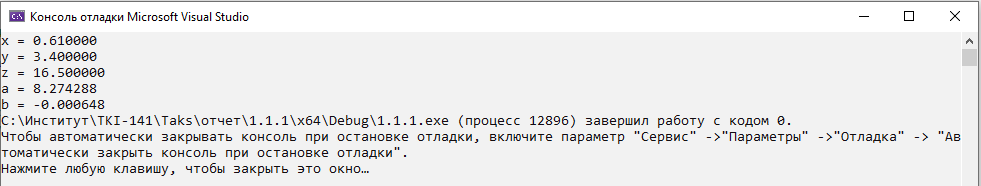


Рисунок  – Результаты выполнения программы

* 1. Выполнение тестовых примеров 1.1

В программе Photomath выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 4, Рисунок 5).



Рисунок  – Результат расчета переменной a

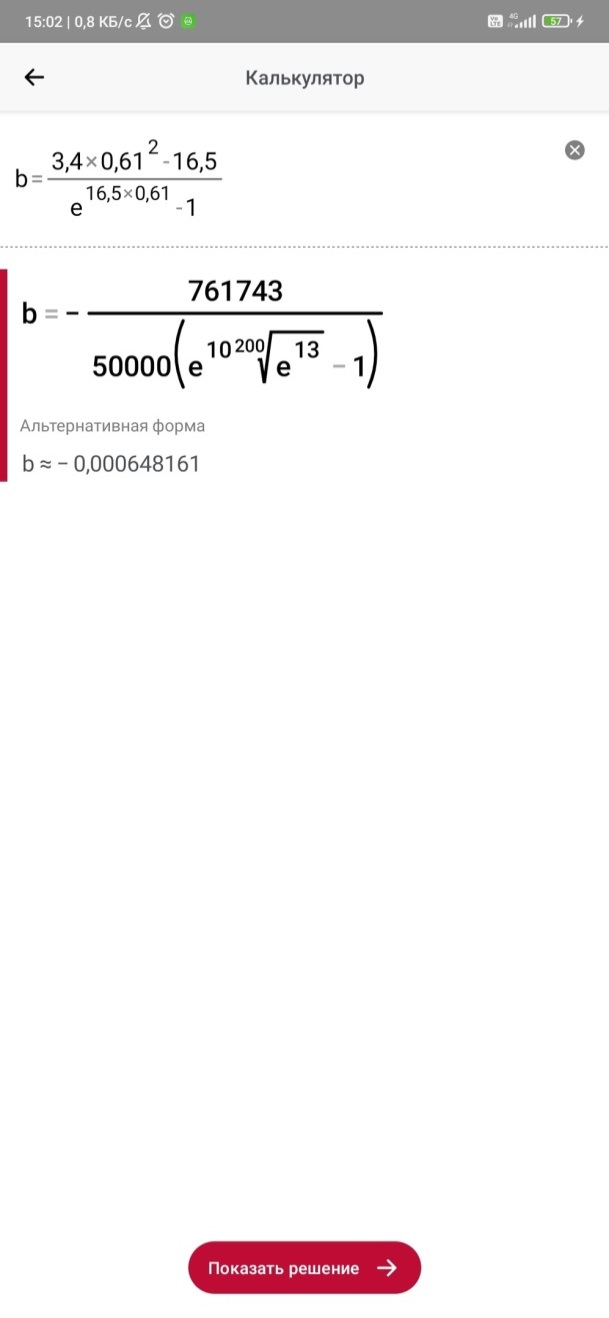


Рисунок  – Результат расчета переменной b

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий 1.1

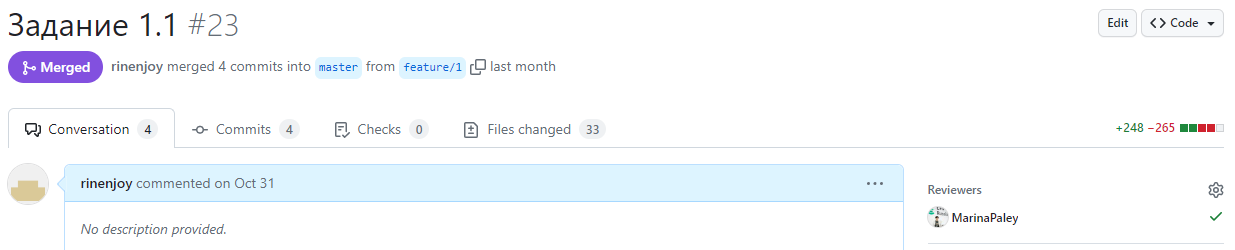


Рисунок 6 - Approved task 1.1

1. Задача 1.2
   1. Формулировка задания 1.2

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 - Исходные данные

|  |  |
| --- | --- |
| Вариант | Задание |
| 6 | Вычислить расстояние между двумя точками с заданными координатами. |

* 1. блок-схема алгоритма 1.2

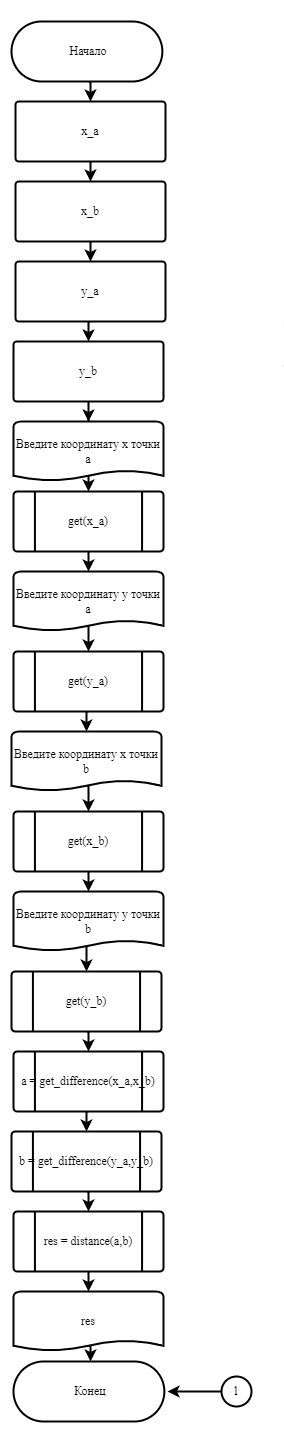
Блок-схема основного алгоритма представлена ниже (Рисунок 7). Блок-схемы используемых функций представлены ниже (Рисунок 8, Рисунок 9) 

Рисунок 7 - Блок-схема основного алгоритма

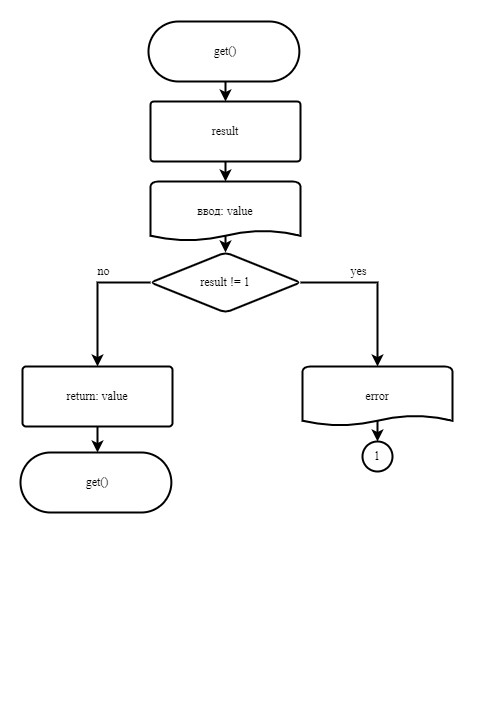


Рисунок 8 - Блок-схема используемой функции

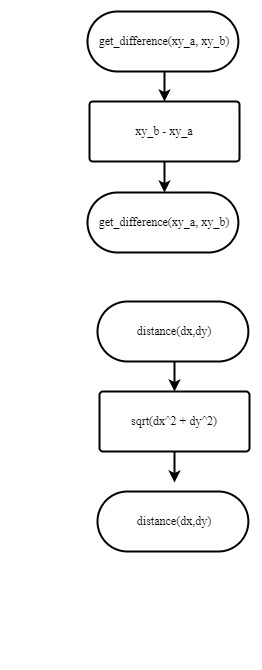


Рисунок 9 - Блок-схема используемой функции

* 1. Текст программы на языке C 1.2

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief Функция получает одну из координат двух точек

\* @return возвращает разницу между координатами

\*/

double get\_difference(double xy\_a, double xy\_b);

/\*\*

\*@brief расчитывает расстояние между двумя заданными точками

\*@param функция ожидает два значения, полученные в результате работы функций get\_x и get\_y

\*@return возвращает длину расстояния между двух точек

\*/

double distance(double dx, double dy);

/\*\*

\*@brief проверяет на правильность пользовательский ввод

\*/

double get();

/\*\*

\*@brief является точкой входа в программу

\*@return возвращает 0, в случае успеха

\*/

int main()

{

puts("Введите координату x точки a: ");

double x\_a = get();

puts("Введите координату y точки a: ");

double y\_a = get();

puts("Введите координату x точки b: ");

double x\_b = get();

puts("Введите координату y точки b: ");

double y\_b = get();

double res = distance(get\_difference(x\_a, x\_b), get\_difference(y\_a, y\_b));

printf\_s("Растояние = %.2lf", res);

return 0;

}

double get()

{

double value;

int result = scanf\_s("%lf", &value);

if (result != 1)

{

puts("error");

abort();

}

return value;

}

double get\_difference(double xy\_a, double xy\_b)

{

return xy\_b - xy\_a;

}

double distance(double dx, double dy)

{

return sqrt(pow(dx, 2) + pow(dy, 2));

}

* 1. Результаты выполнения программы 1.2

Результаты выполнения программы представлены ниже (Рисунок 10, Рисунок 11, Рисунок 12).

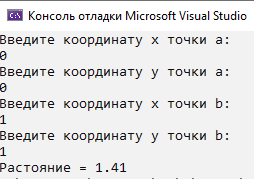


Рисунок 10 - Результаты выполнения программы

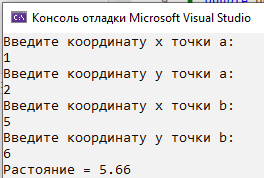


Рисунок 11 - Результат выполнения программы

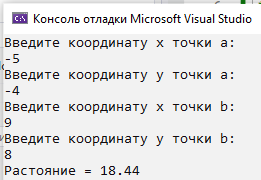


Рисунок 12 - Результат выполнения программы

* 1. Выполнение тестовых примеров 1.2

На сайте onlinemschool.com выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 13, Рисунок 14).

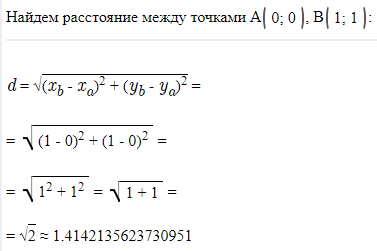


Рисунок 13 – Тестовый расчет

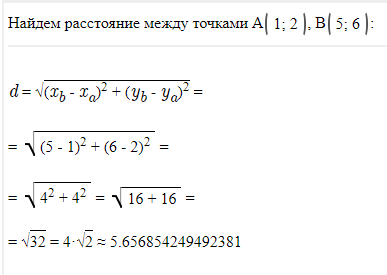


Рисунок 14 – Тестовый расчет

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий 1.2

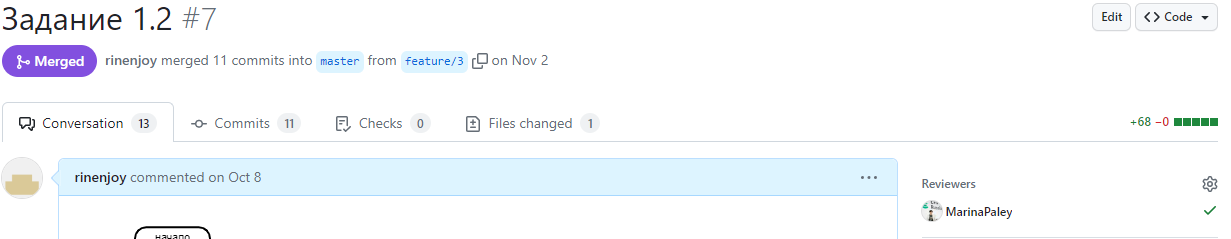


Рисунок 15 - Approved task 1.2

1. задача 1.3
   1. Формулировка задания 1.3

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 3 - Исходные данные

|  |  |
| --- | --- |
| Вариант | Задание |
| 6 | Рассчитайте давление, производимое бетонной плитой, масса которой равна *m* кг, а площадь опоры *S* м2. |

* 1. Блок-схема алгоритма 1.3

Блок-схема основного алгоритма представлена ниже (Рисунок 16). Блок-схемы используемых функций представлены ниже (Рисунок 17, Рисунок 18).

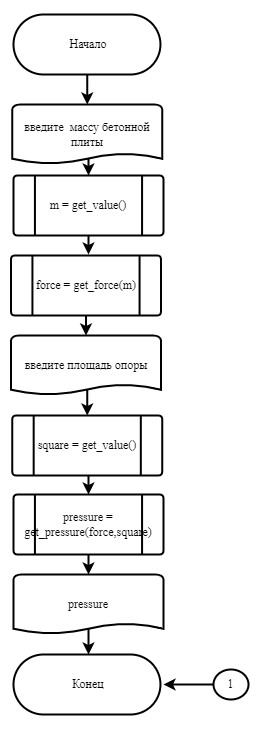


Рисунок 16 - Блок-схема основного алгоритма

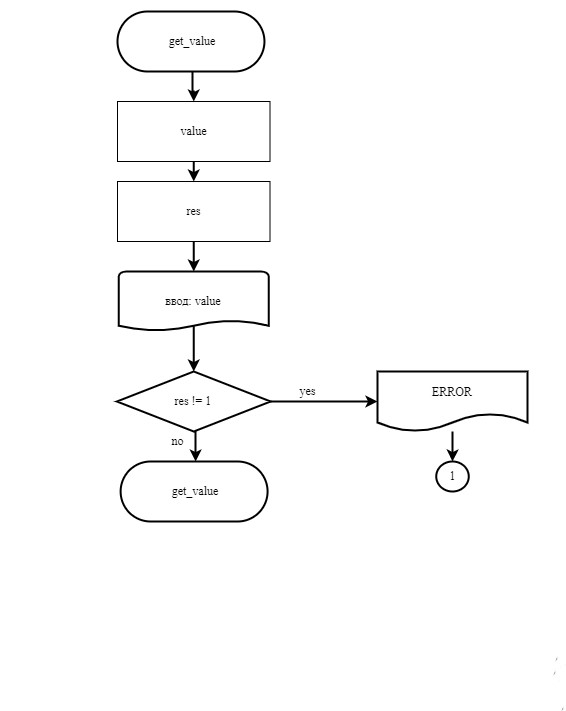


Рисунок 17 - Блок-схема используемой функции

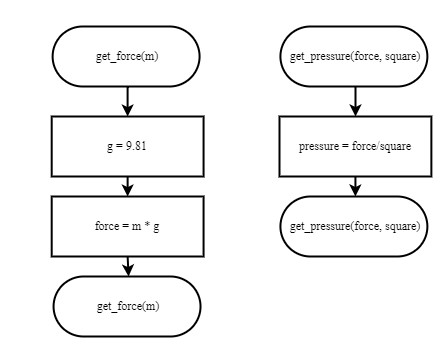


Рисунок 18 - Блок-схемы используемых функций

* 1. Текст программы на языке C 1.3

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

/\*

\* @brief Функция получет пользовательское значение

\*/

double get\_value();

/\*

\* @brief Функция вычисляет силу force

\* @param m - масса бетонной плиты [кг]

\* @return сила force [Н]

\*/

double get\_force(double m);

/\*

\* @brief Функция вычисляет давление, производимое бетонной плитой

\* @param force - сила давления бетонной плиты

\* @param square - площадь опоры

\* @return давление pressure [Па]

\*/

double get\_pressure(double force, double square);

/\*

\* @brief является точкой входа в программу

\* @return 0, в случае успеха

\*/

int main() {

puts("Введите массу (m) бетонной плиты [кг]: ");

double m = get\_value();

puts("Введите площадь опоры [кв.м] : ");

printf\_s("Давление, производимое бетонной плитой = %.2lf [Па]", get\_pressure(get\_force(m), get\_value()));

return 0;

}

double get\_value()

{

double value;

double res = scanf\_s("%lf", &value);

if (res != 1)

{

puts("Error");

abort();

}

return value;

}

double get\_force(double m)

{

const double g = 9.81;

return (m \* g);

}

double get\_pressure(double force, double square)

{

return (force / square);

}

* 1. Результаты выполнения программы 1.3

Результаты выполнения программы представлены ниже (Рисунок 19, Рисунок 20).

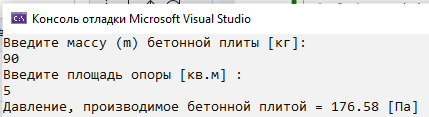


Рисунок 19 - Результаты выполнения программы

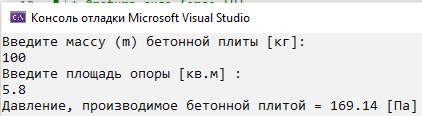


Рисунок 20 - Результаты выполнения программы

* 1. Выполнение тестовых примеров 1.3

На сайте planetcalc.ru выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 21, Рисунок 22).

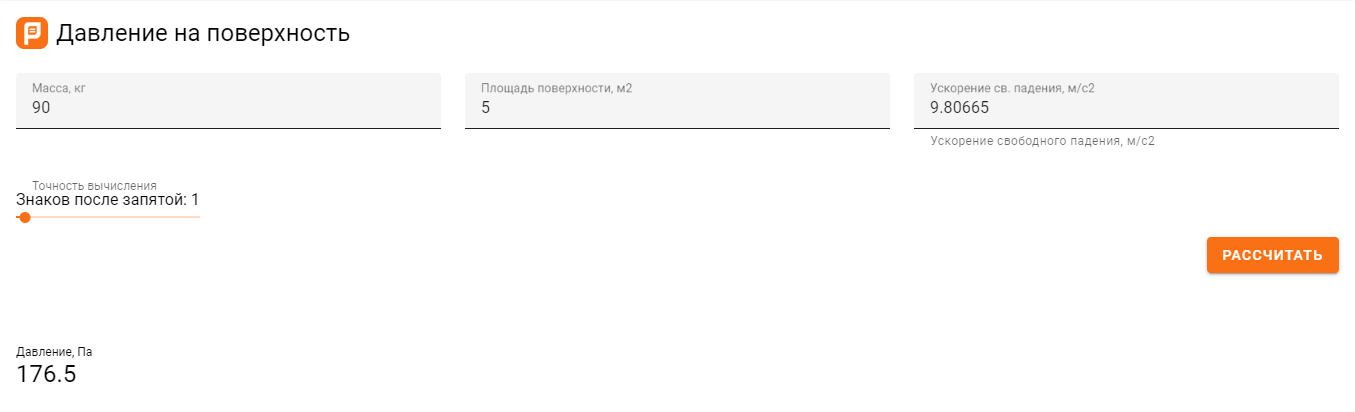


Рисунок 21 – Тестовый расчет

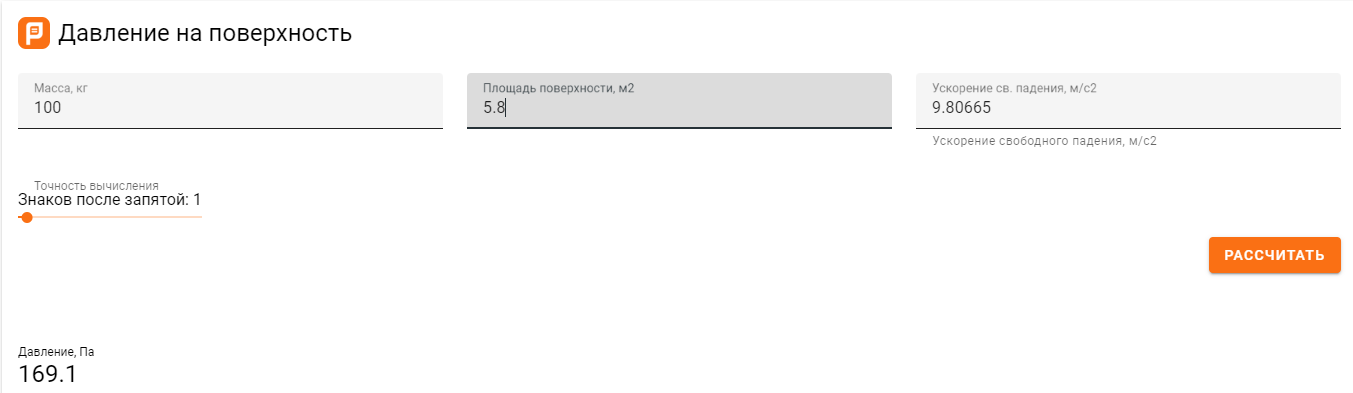


Рисунок 22 – Тестовый расчет

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий 1.3

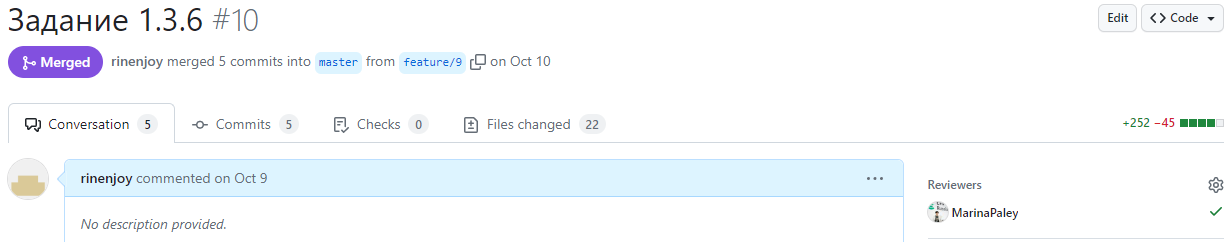


Рисунок 23 - Approved task 1.3