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

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

По дисциплине «Языки программирования»

ВАРИАНТ 9

Выполнил: ст. гр. ТКИ-141

Ткачев Вадим Евгеньевич

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

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

Москва 2024

Оглавление

[1 Решение задачи 2–1 3](#_Toc182509025)

[1.1 Формулировка задачи 2–1 3](#_Toc182509026)

[1.2 Блок-схема алгоритма задание 2–1 4](#_Toc182509027)

[1.4 Результаты выполнения программы 2–1 9](#_Toc182509028)

[1.5 Выполнение тестовых примеров задание 2–1 10](#_Toc182509029)

[1.6 Отметка о выполнении задания в веб-хостинге системы контроля версий задание 2–1 12](#_Toc182509030)

[2 Решение задачи 2–2 13](#_Toc182509031)

[2.1 Формулировка задачи 2–2 13](#_Toc182509032)

[2.2 Блок-схема алгоритма задание 2–2 14](#_Toc182509033)

[2.3 Текст программы на языке С задание 2–2 15](#_Toc182509034)

[2.4 Результаты выполнения программы 2–2 16](#_Toc182509035)

[2.5 Выполнение тестовых примеров задание 2–2 17](#_Toc182509036)

[2.6 Отметка о выполнении задания в веб-хостинге системы контроля версий задание 2–2 18](#_Toc182509037)

[3 Решение задачи 2–3 19](#_Toc182509038)

[3.1 Формулировка задачи 2–3 19](#_Toc182509039)

[3.2 Блок-схема алгоритма задание 2–3 20](#_Toc182509040)

[3.3 Текст программы на языке С задание 2–3 21](#_Toc182509041)

[3.4 Результаты выполнения программы 2–3 22](#_Toc182509042)

[3.5 Выполнение тестовых примеров задание 2–3 24](#_Toc182509043)

[3.6 Отметка о выполнении задания в веб-хостинге системы контроля версий задание 2–3 27](#_Toc182509044)

1. Решение задачи 2–1
   1. Формулировка задачи 2–1

Создать консольное приложение для решения задачи, представленной в таблице, с использованием перечислимого типа. Выбор формулы вычисления зависит от пользователя. Данные для решения задачи так же вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 1 – Исходные данные

|  |  |
| --- | --- |
| Вариант | Задачи |
| 9 | Заданы длина, ширина и высота параллелепипеда. Определить его:   * объём; * площадь поверхности. |

* 1. Блок-схема алгоритма задание 2–1

Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы функций get\_volume, get\_surface\_area (Рисунок 2), input (Рисунок 3), input\_of\_integer (Рисунок 4), get\_above\_zero\_value (Рисунок 5) представлены ниже.

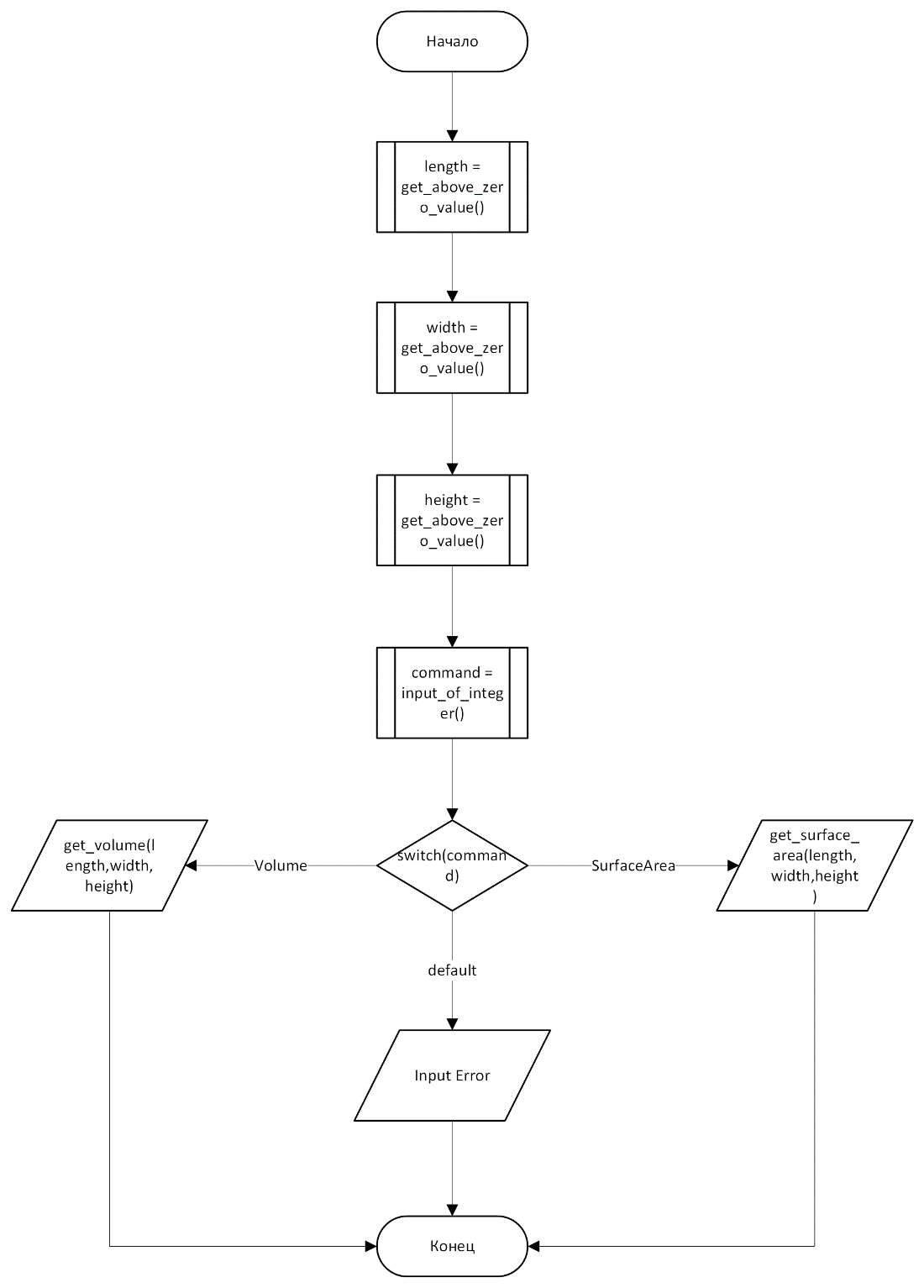


Рисунок 1 – Блок-схема основного алгоритма

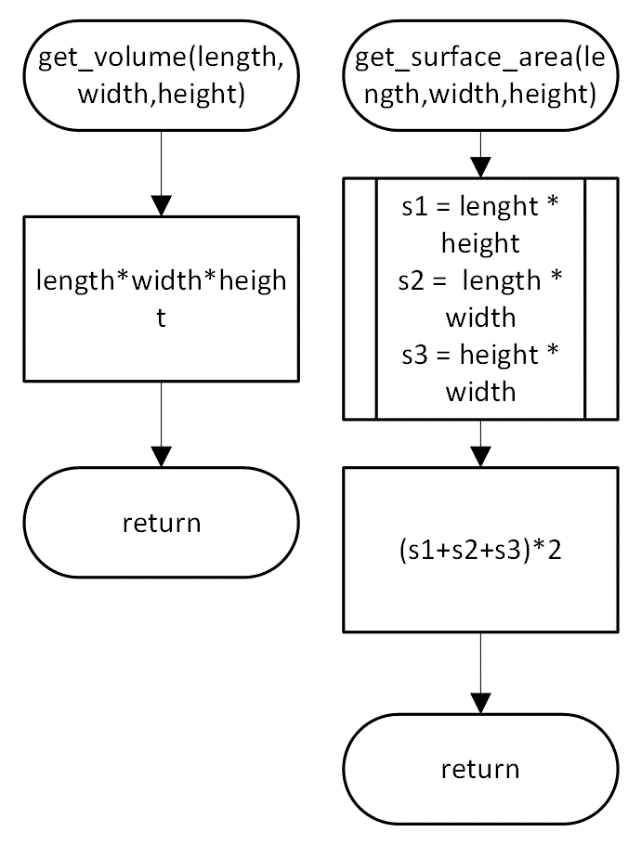


Рисунок 2 – Блок-схема функций get\_volume, get\_surface\_area

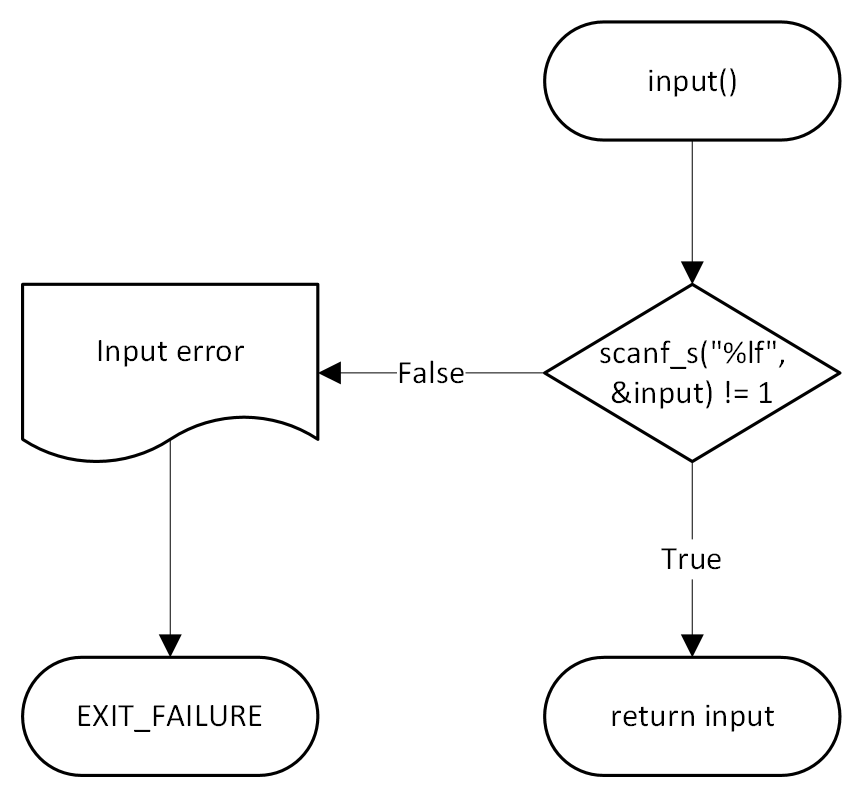


Рисунок 3 – Блок-схема проверки на ввод

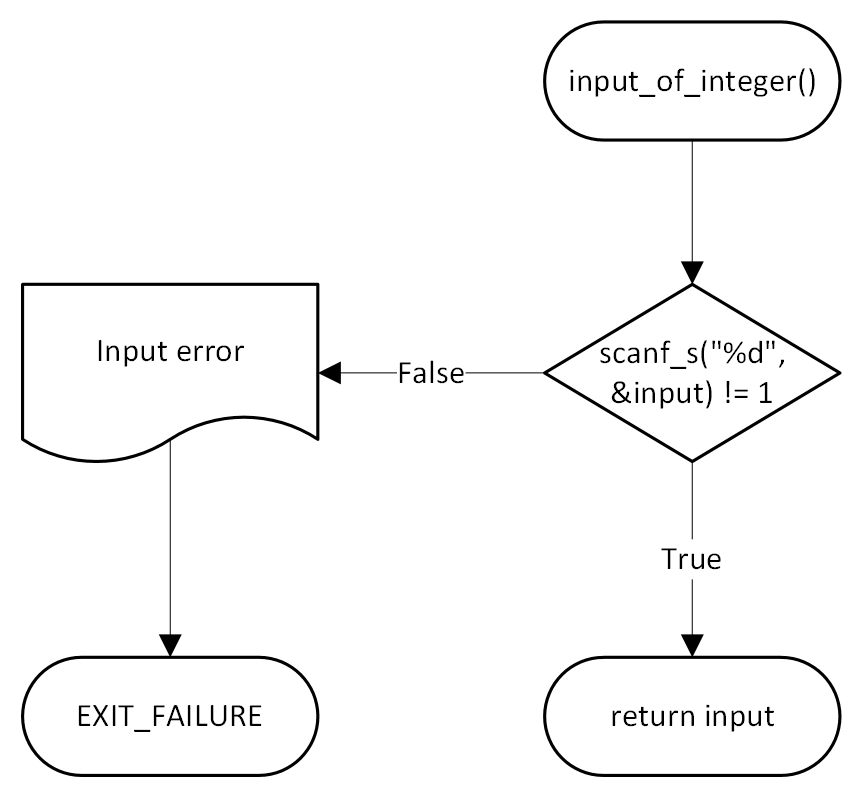


Рисунок 4 – Блок-схема функции проверки на целое число

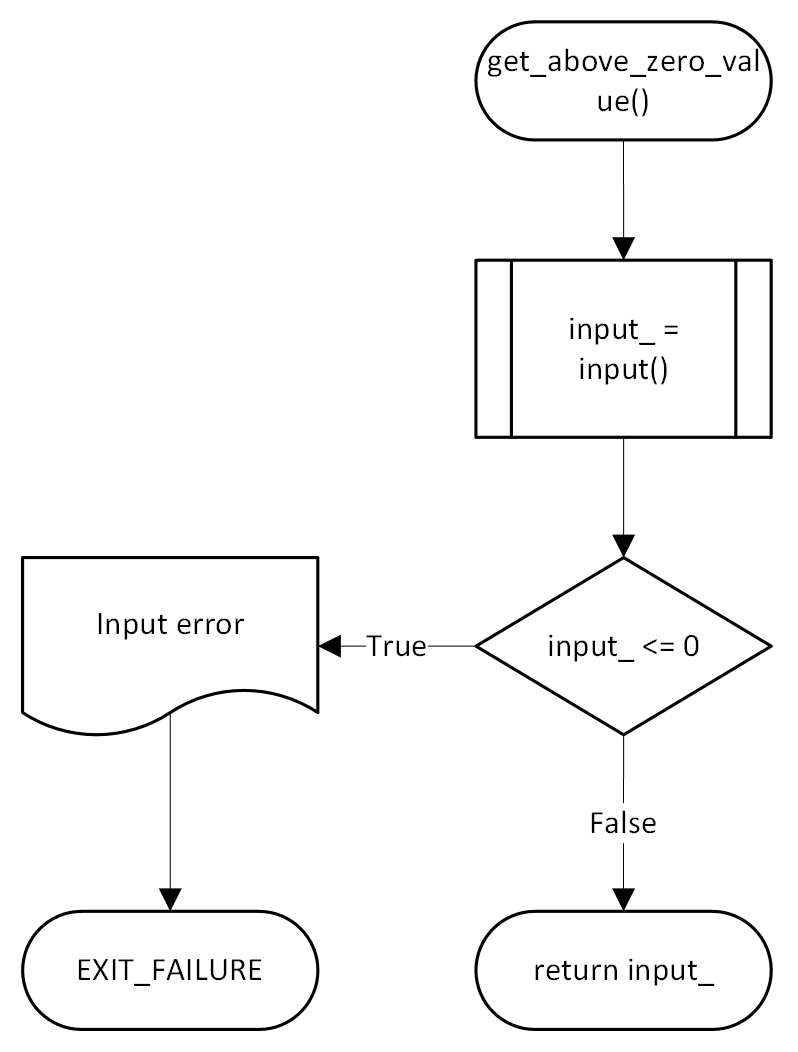


Рисунок 5 – Блок-схема функции проверки значение на ноль

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief Функция расчета объема

\* @param length - значение длины

\* @param width - значение ширины

\* @param height - значение высоты

\* @return Вывод объема

\*/

double get\_volume(double length, double width, double height);

/\*\*

\* @brief Функция расчета площади

\* @param length - значение длины

\* @param width - значение ширины

\* @param height - значение высоты

\* @return Выводит итоговую площадь

\*/

double get\_surface\_area(double length, double width, double height);

/\*\*

\* @brief Функция проверки введенных значений.

\* @remarks При неправильном вводе программа будет закрыта с кодом ошибки /c EXIT\_FAILURE

\* @return возвращает значение, если выполнено успешно, или ошибку, если иначе

\*/

double input(void);

/\*\*

\* @brief Функция проверяет на 0

\* @remarks При неправильном вводе программа будет закрыта с кодом ошибки /c EXIT\_FAILURE

\* @return Возвращает ошибку если число 0 или меньше, или значение если все хорошо

\*/

double get\_above\_zero\_value(void);

/\*\*

\* @brief Функция проверки введенного целого числа

\* @remarks При неправильном вводе программа будет закрыта с кодом ошибки /c EXIT\_FAILURE

\* @return Возвращает ошибку если число не integer

\*/

int input\_of\_integer(void);

enum Choice

{

Volume = 1,

SurfaceArea = 2

};

/\*\*

\* @brief Точка входа в функцию

\* @return Возвращает код ошибки

\*/

int main(void)

{

printf("Enter lenght,width and height of cube\n");

double length = get\_above\_zero\_value(), width = get\_above\_zero\_value(), height = get\_above\_zero\_value();

printf("enter the command\n1 - calculate volume of the cube\n2 - calculate surface area of cube\n");

int command = input\_of\_integer();

switch((enum Choice) command)

{

case Volume: printf("%lf",get\_volume(length, width, height)); break;

case SurfaceArea: printf("%lf", get\_surface\_area(length, width, height)); break;

default: puts("Not command/Input error");

}

return 0;

}

int input\_of\_integer(void)

{

int input;

if (scanf\_s("%d", &input) != 1)

{

printf("Input error");

exit(EXIT\_FAILURE);

}

return input;

}

double input(void)

{

double input;

if (scanf\_s("%lf", &input) != 1)

{

printf("Input error");

exit(EXIT\_FAILURE);

}

return input;

}

double get\_volume(double length, double width, double height)

{

return length \* width \* height;

}

double get\_surface\_area(double length, double height, double width)

{

const double s1 = length \* height, s2 = length \* width, s3 = height \* width;

return (s1 + s2 + s3) \* 2;

}

double get\_above\_zero\_value(void)

{

double input\_ = input();

if (input\_ <= 0)

{

printf("Input error");

exit(EXIT\_FAILURE);

}

return input\_;

}

* 1. Результаты выполнения программы 2–1

Результаты выполнения программы в C представлена ниже (Рисунок 6, Рисунок 7)

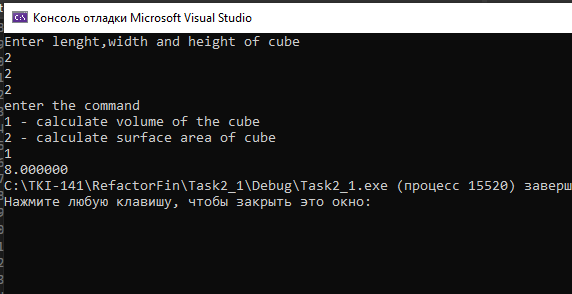


Рисунок 6 – Вывод объема куба

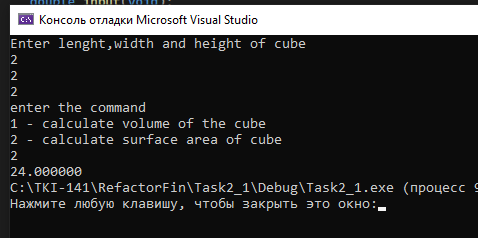


Рисунок 7 – Вывод площади куба

* 1. Выполнение тестовых примеров задание 2–1

В программе Pythonвыполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 8, Рисунок 9)

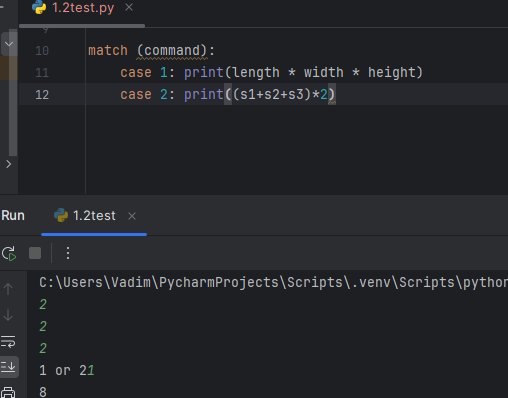


Рисунок 8 – Тестовое решение: вывод объема куба

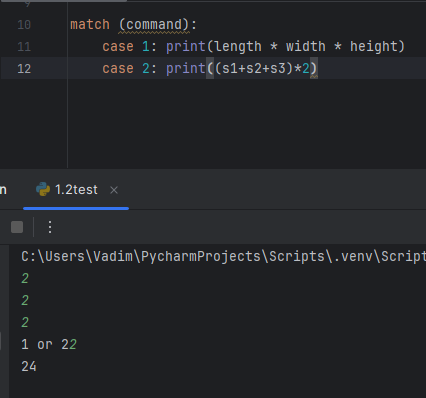


Рисунок 9 – Тестовое решение: вывод площади куба

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий задание 2–1

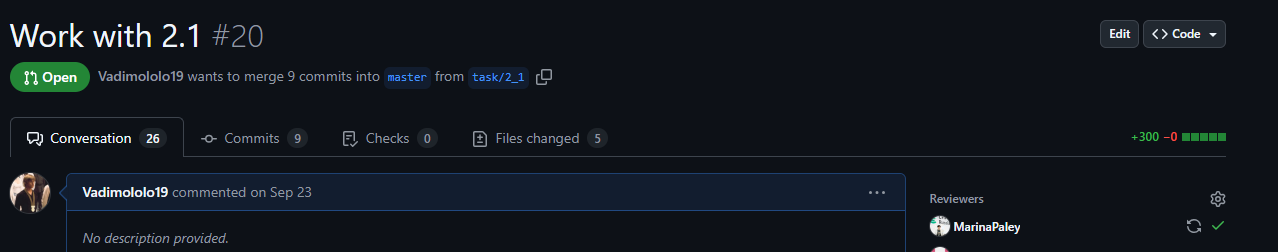
Ниже представлено доказательство того, что задание 2–1 было принято. (Рисунок 10)

Рисунок 10 – Approve задания 2.1

1. Решение задачи 2–2
   1. Формулировка задачи 2–2

Создать консольное приложение с организацией разветвлённого процесса для решения заданной в таблице функции. Данные для решения вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 1 – Исходные данные

|  |  |  |
| --- | --- | --- |
| Вариант | Функция | Константы |
| 9 |  |  |

* 1. Блок-схема алгоритма задание 2–2

Блок-схема основного алгоритма представлена ниже (Рисунок 11). Блок-схемы функций calculate\_X1 и calculate\_x2 (Рисунок 12) и проверки на ввод (Рисунок 13) представлены ниже.

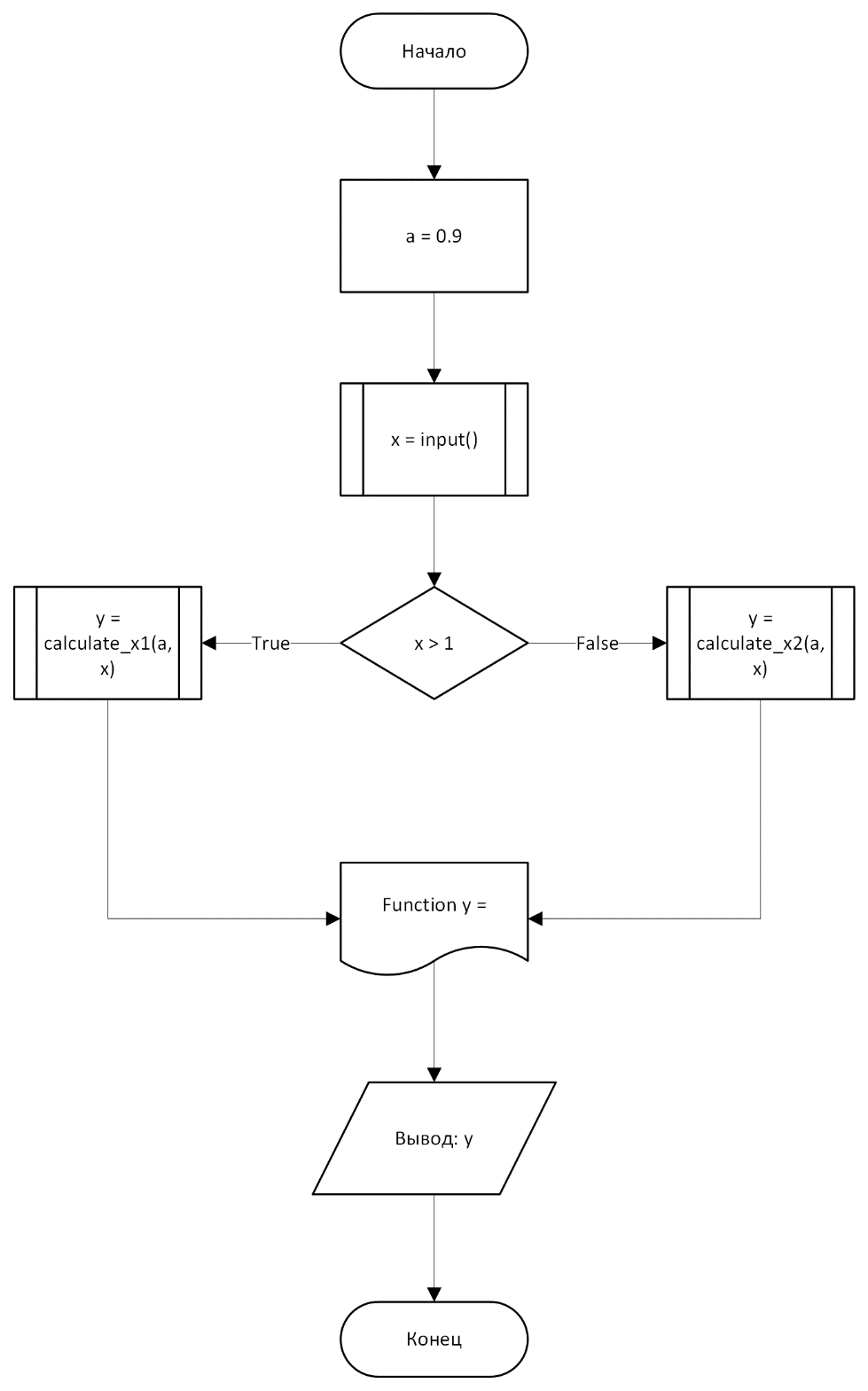


Рисунок 11 –­ Блок-схема основного алгоритма

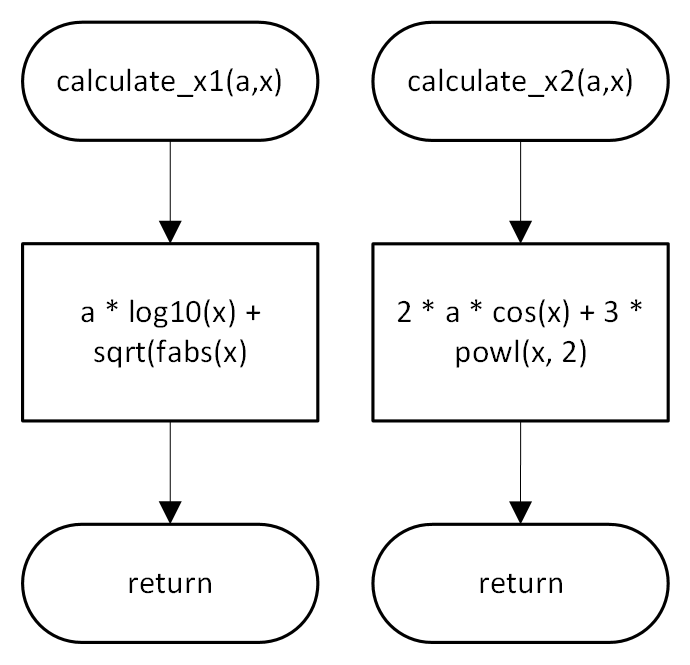


Рисунок 12 – Блок-схема функций calculate\_x1 и calculate\_x2

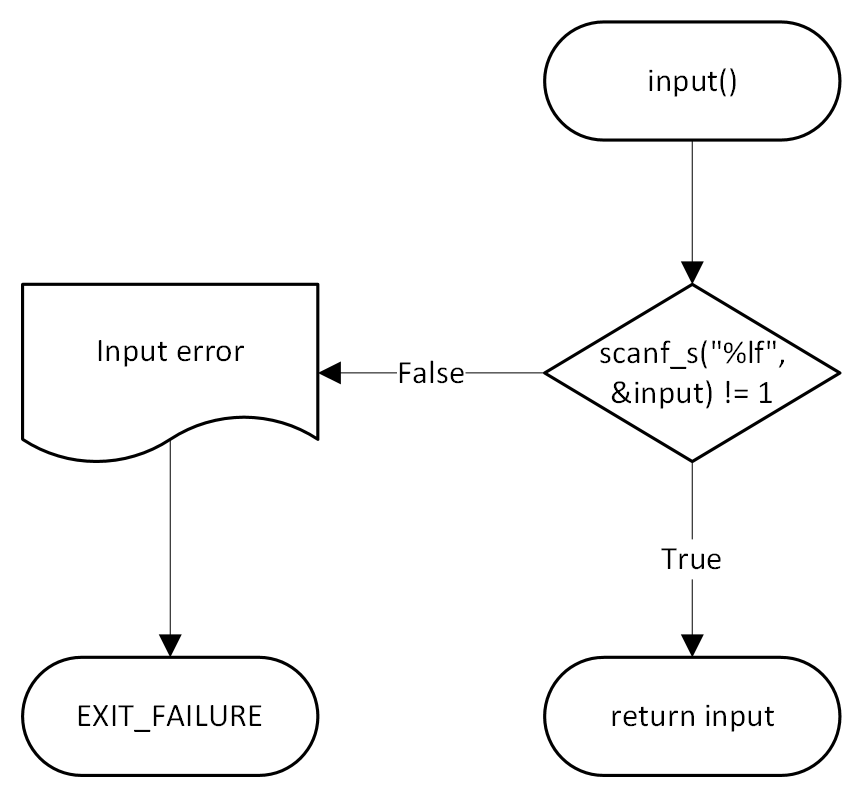


Рисунок 13 – Блок-схема проверки на ввод

* 1. Текст программы на языке С задание 2–2

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief Функция проверки введенных значений.

\* @remarks При неправильном вводе программа будет закрыта с кодом ошибки /c EXIT\_FAILURE

\* @return возвращает значение, если выполнено успешно, или ошибку, если иначе.

\*/

double input();

/\*\*

\* @brief Функция счета функции при условиях.

\* @return Вовзращает значение функции если x > 1.

\*/

double calculate\_x1(double a,double x);

/\*\*

\* @brief Функция счета функции при условиях.

\* @return Вовзращает значение функции если x <= 1.

\*/

double calculate\_x2(double a, double x);

/\*\*

\* @brief Точка входа.

\* @return Возвращает с кодом ошибки.

\*/

int main()

{

const double a = 0.9;

double x = input(), y;

if (x > 1)

{

y = calculate\_x1(a, x);

}

else

{

y = calculate\_x2(a, x);

}

printf("Function y = %lf", y);

return 0;

}

double calculate\_x1(double a,double x)

{

return a \* log10(x) + sqrt(fabs(x));

}

double calculate\_x2(double a,double x)

{

return 2 \* a \* cos(x) + 3 \* powl(x, 2);

}

double input()

{

double input;

if (scanf\_s("%lf", &input) != 1)

{

printf("Input error");

exit(EXIT\_FAILURE);

}

return input;

}

* 1. Результаты выполнения программы 2–2

Результаты выполнения программы в C представлена ниже (Рисунок 14, Рисунок 15)

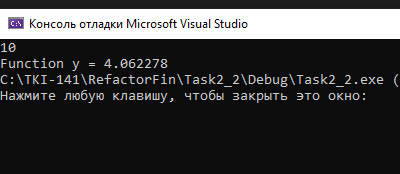


Рисунок 14 – Результаты выполнения программы если x > 1

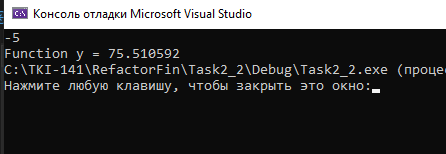


Рисунок 15 – Результаты выполнения программы если x ≤ 1

* 1. Выполнение тестовых примеров задание 2–2

В программе Pythonвыполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 16, Рисунок 17).

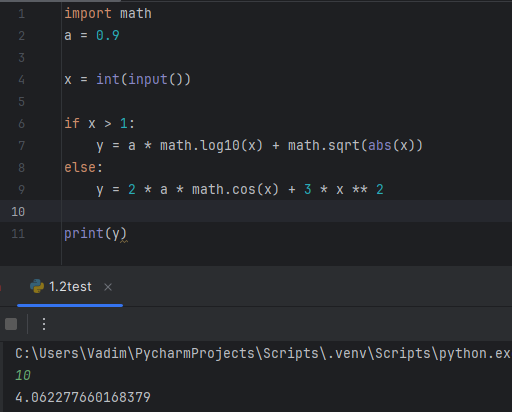


Рисунок 16 – Результат расчета при x > 1

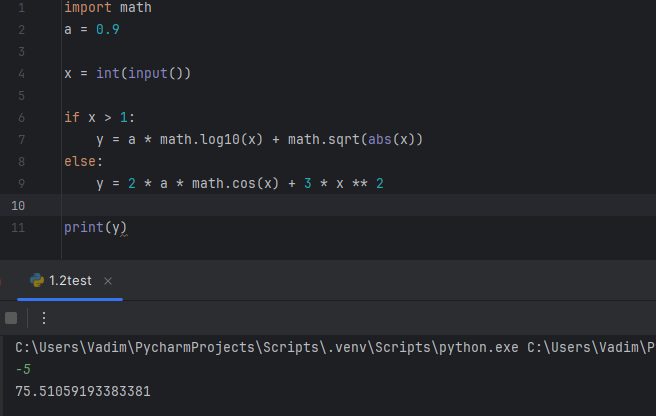


Рисунок 17 – Результат расчёта при x ≤ 1

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий задание 2–2

Ниже представлено доказательство того, что задание 2–2 было принято. (Рисунок 18)

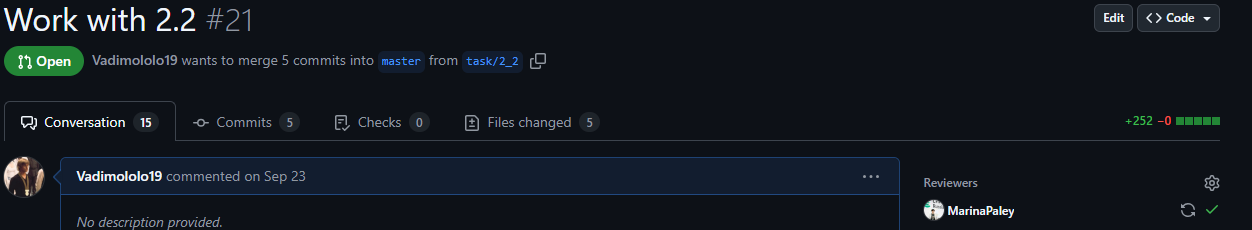


Рисунок 18 – Approve задачи 2–2

1. Решение задачи 2–3
   1. Формулировка задачи 2–3

Создать консольное приложение с организацией разветвлённого процесса для решения задачи из таблицы. Данные для решения вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 1 – Исходные данные

|  |  |
| --- | --- |
| Вариант | Задачи |
| 9 | Записать программу, которая на ввод времени суток выводит соответствующее пожелание доброго утра, доброго дня, доброго вечера и спокойной ночи. |

* 1. Блок-схема алгоритма задание 2–3

Блок-схема основного алгоритма представлена ниже (Рисунок 19). Блок-схема функции input (Рисунок 20) представлена ниже.

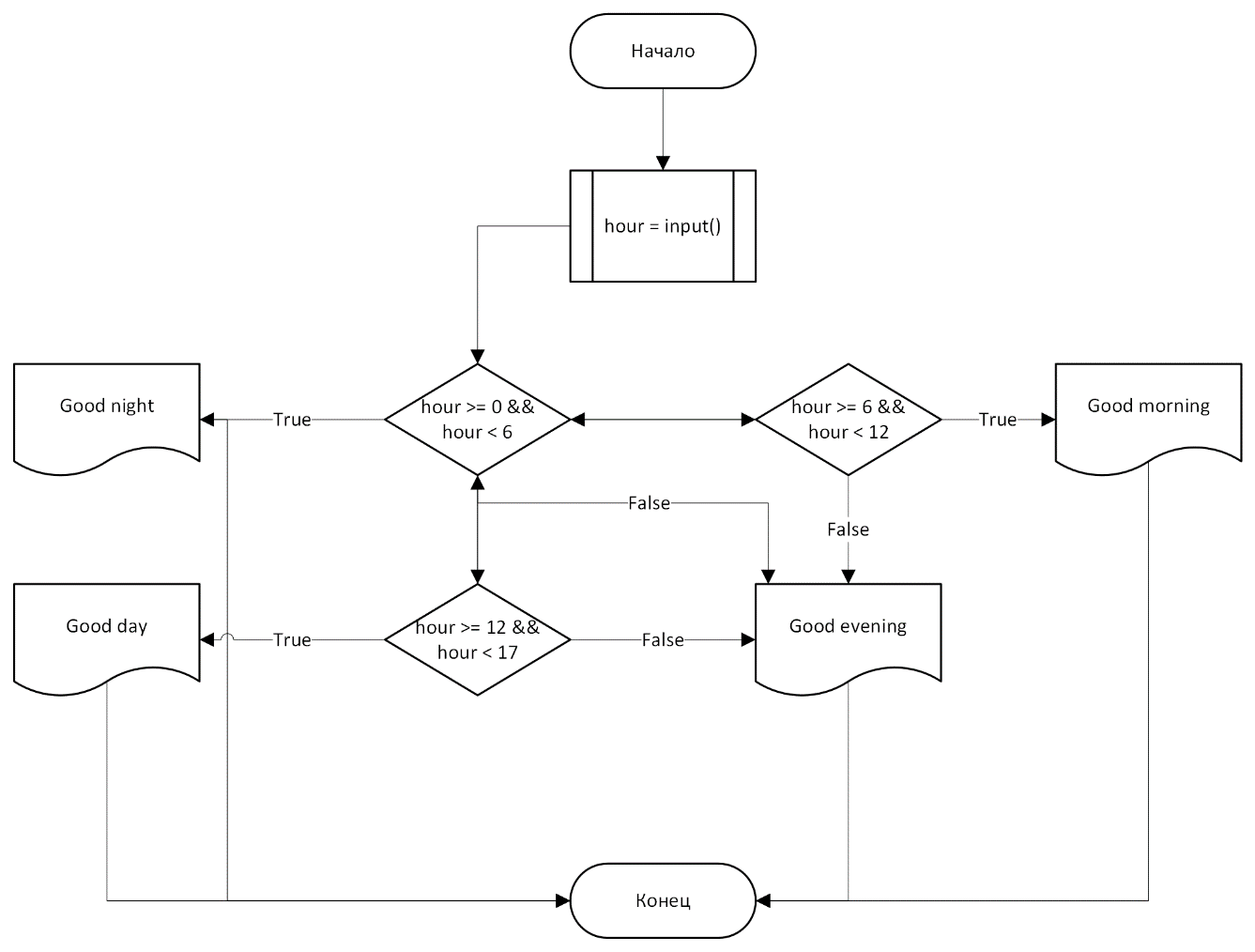


Рисунок 19 ­– Блок-схема основного алгоритма

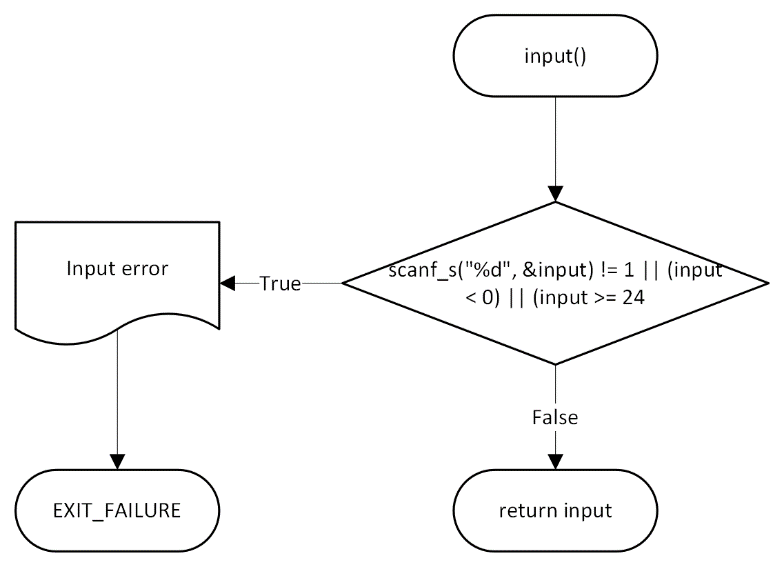


Рисунок 20 – Блок-схема функции проверки ввода

* 1. Текст программы на языке С задание 2–3

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief Функция проверки введенных значений.

\* @return возвращает значение, если выполнено успешно, или ошибку, если иначе

\*/

int input();

/\*\*

\* @brief Точка входа

\* @return Возвращает результат врограммы

\*/

int main()

{

printf("Your hour?\n");

int hour = input();

if (hour >= 0 && hour < 6)

{

printf("Good night");

}

else if (hour >= 6 && hour < 12)

{

printf("Good morning");

}

else if (hour >= 12 && hour < 17)

{

printf("Good day");

}

else

{

printf("Good evening");

}

return 0;

}

int input()

{

int input;

if (scanf\_s("%d", &input) != 1 || (input < 0) || (input >= 24))

{

printf("Input error");

exit(EXIT\_FAILURE);

}

return input;

}

* 1. Результаты выполнения программы 2–3

Результаты выполнения программы в C представлена ниже (Рисунок 21,Рисунок 22, Рисунок 23, Рисунок 24)

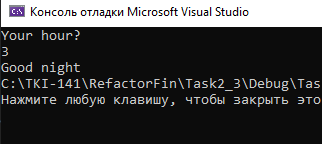


Рисунок 21 – Результат выполнения программы если по часам ночь

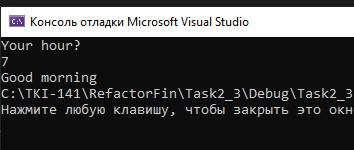


Рисунок 22 – Результат выполнения программы если по часам утро

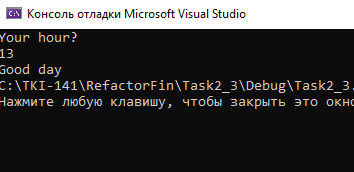


Рисунок 23 – Результат выполнения программы если по часам день

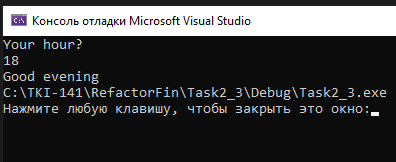


Рисунок 24 – Результат выполнения программы если по часам вечер

* 1. Выполнение тестовых примеров задание 2–3

В программе Pythonвыполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 25, Рисунок 26, Рисунок 27,Рисунок 28).

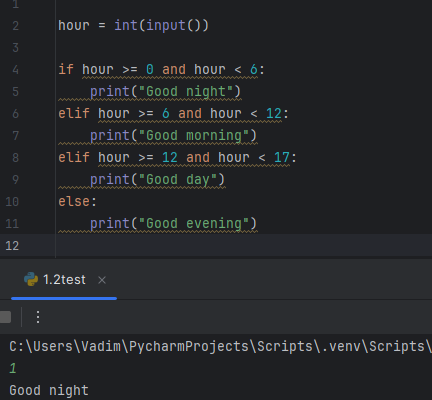


Рисунок 25 – Результат программы если по часам ночь

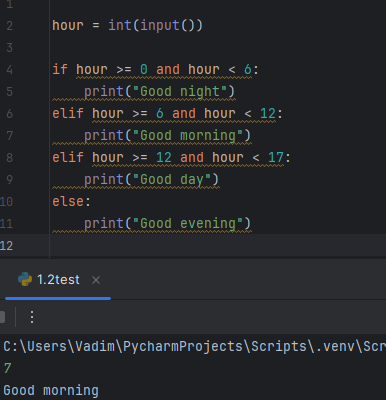


Рисунок 26 – Результат программы если по часам утро

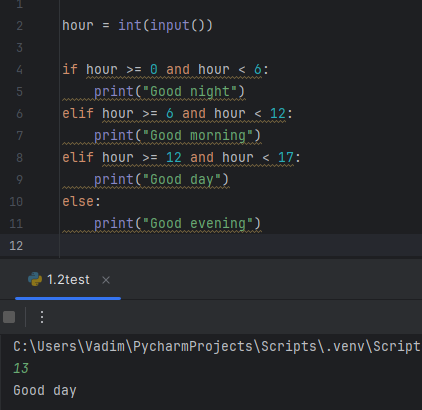


Рисунок 27 – Результат программы если по часам день

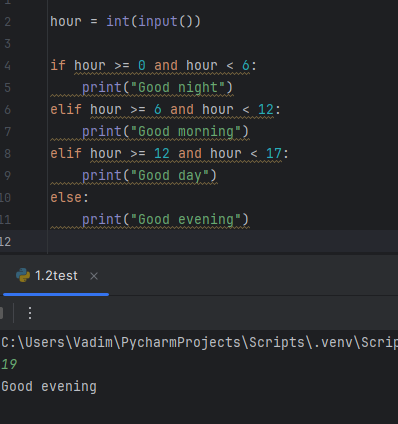


Рисунок 28 – Результат программы если по часам вечер

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий задание 2–3

Ниже представлено доказательство того, что задание 2–3 было принято. Рисунок 29)

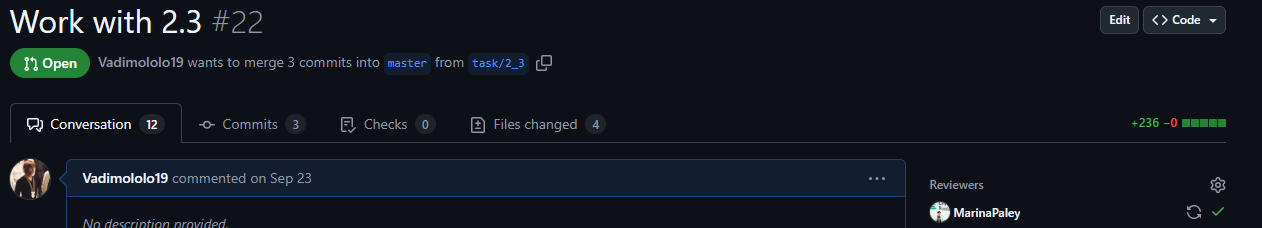


Рисунок 29 – Approve задачи 2–3