МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

#### Факультет экономики, менеджмента и информационных технологий

(факультет)

### Кафедра Систем управления и информационных технологий в строительств

# ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине Объектно-ориентированное программирование

Тема «Базовые конструкции языка С++. Управляющие конструкции»

Выполнил студент группы ИСТ-214 Д. В. Тюленев Подпись, дата Инициалы, фамилия

## Принял Е. Н. Королев Подпись, дата Инициалы, фамилия

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

2022

Лабораторная работа № 1

«Базовые конструкции языка С++. Управляющие конструкции»

Вариант №4

1. Цель работы: Научиться работать с базовыми конструкциями языка С++ и управляющими конструкциями.
2. Задания на лабораторную работу:
   1. Найти значение алгебраического выражения. Вывести результаты на печать. Все результаты выводить в развёрнутом виде (например: «Сумма чисел А и B равна 3.7854»). Значения вводимых величин должны иметь не менее четырёх значащих цифр и задаются студентом самостоятельно.

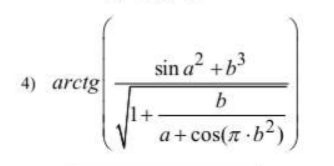


Рисунок 1 － Алгебраическое выражение

* 1. Составить программу вычисления функции F(x), разложенной в ряд, сходящийся в заданной области. Вычисления проводить до тех пор, пока модуль разности между последующим и предыдущим членами ряда не будет меньше или равен ε. Значение ε задаётся константой и составляет 0,001. Решить задачу для различных значений х. При решении задач массивы не использовать. Специальные функции для возведения в степень не использовать.

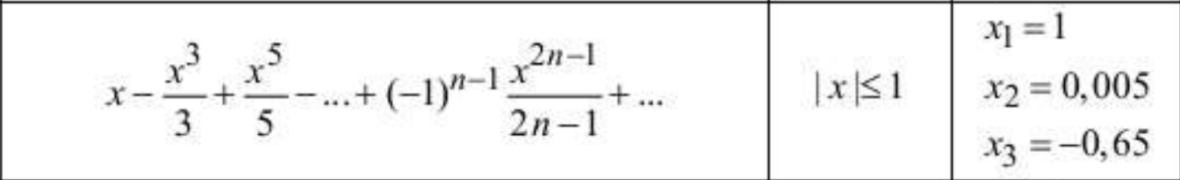


Рисунок 2 － Функция F(x)

* 1. Используя средства отладчика среды, обнаружить все ошибки и сделать программу работоспособной. Для отчёта по данному заданию необходимо: показать работающие программы; код программ должен быть структурированным; к каждой программе необходимо составить список обнаруженных ошибок (согласно сообщениям отладчика), а также пояснения того, как вы исправляли эти ошибки.
  2. Написать программу проверки на положительность введённого числа.
  3. С помощью оператора выбора реализовать меню.

1. Ход выполнения:
   1. Задание №1:

Определим и реализуем функцию для решения задания, которая возвращает значение типа «bool», определяющее завершилось задание с ошибкой или нет. Установим фиксированное количество знаков после запятой (4 разряда) при помощи манипулятора потока «fixed» (указываем, что устанавливаемая точность относиться к количеству знаков после запятой), и метода «precision» класса «ostream». После объявляем две переменные «a, b» вещественного типа double. Пригласим к вводу пользователя и через объект «cin» считываем из потока данных значение в переменные. Определим реализацию функции «task::function», которая в качестве параметров принимает две переменные и возвращает посчитанное значение выражения. Под конец выведем результат.

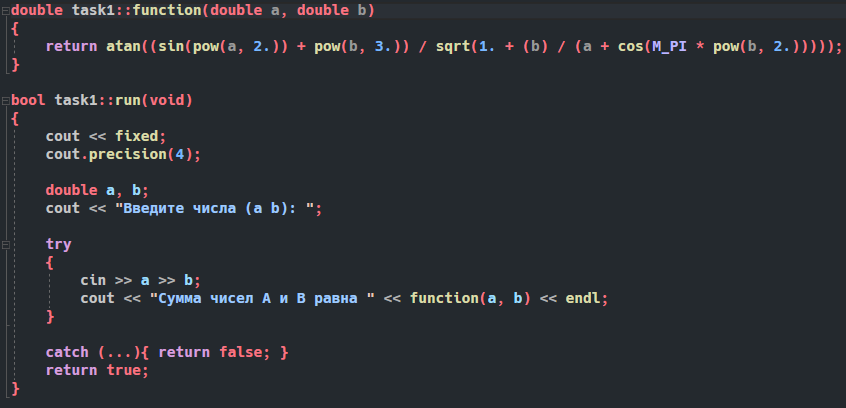


Рисунок 3.1 － Решение задания №1

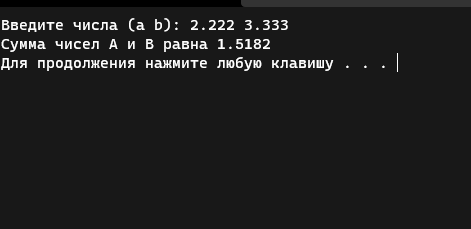


Рисунок 3.2 － Вывод задания №1

* 1. Задание №2:

Определим и реализуем функцию для решения задания, которая возвращает значение типа «bool», определяющее завершилось задание с ошибкой или нет при помощи конструкции «try-catch». Определим переменную «x» вещественного типа double для хранения аргумента функции. Приглашаем на ввод пользователя и считываем значение из потока в переменную.

По условию задания область сходимости равна: . Поэтому необходимо добавить проверку введённого значения: если значение не принадлежит отрезку, то выводим сообщение об ошибке и генерируем исключение.

После выводим результат, посчитанный при помощи вызова функции «task2::function».

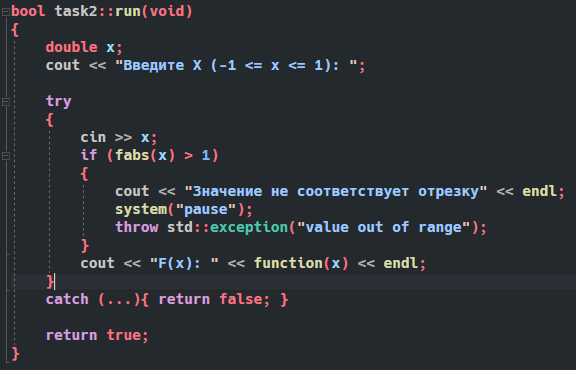


Рисунок 4.1 － Решение задания №2

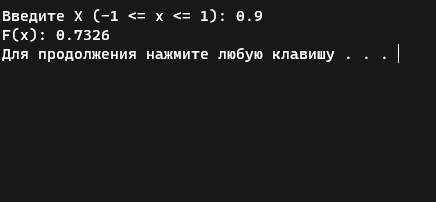


Рисунок 4.2 － Вывод задания №2

Функция «task2::function» производит вычисление значения разложенной в ряд функции F(x). Она принимает в качестве параметра значение аргумента функции F(x) и возвращает рассчитанную сумму ряда.

Инициализируем переменные «current\_state» (значение прошлой итерации), «iter» (значение текущей итерации), «sum» (сумма ряда). После открываем цикл в интервале с шагом 1. В теле цикла высчитываем значение текущего члена ряда и записываем результат в переменную «iter»; складываем значение текущей посчитанной суммы ряда с найденным новым значением члена и перезаписываем переменную «sum». После производим проверку содержит ли переменная «current\_state» значение отличное от нуля. Если утверждение истинно, то в теле условия производим сравнение модуля разности между последующим и предыдущим членами с константой «EBSILON» (#define EBSILON 0.001), и если модуль будет меньше или равен значению константы: происходит выход из цикла. В конце итерации устанавливаем новое значение для переменной «current\_state». В конце тела функции с помощью оператора «return» возвращаем значение посчитанной суммы ряда функции.

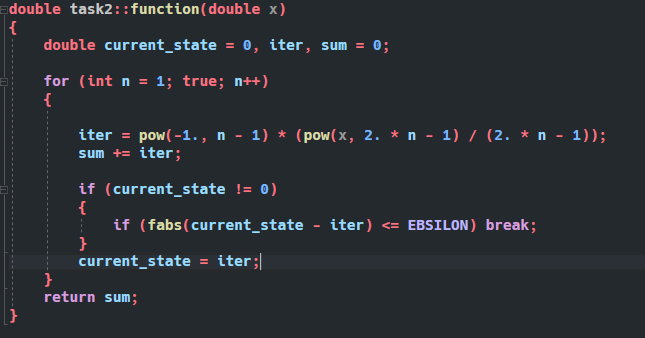


Рисунок 5 － Функция вычисления значения ряда

* 1. Задание №3:

Для запуска подзадач данного задания реализованна функция «task3::run» которая не принимает аргументов и возвращает значение типа «bool» (определяет завершилось задание с ошибкой или нет).

В начале инициализируется массив состоящий из указателей на функции отвечающих за подзадачи. Далее производится ввод значение номера задания со стороны пользователя, происходит проверка на валидность введённого числа. После в соответствии с номером функции определяется индекс элемента массива, извлекается необходимый указатель и вызывается функция.

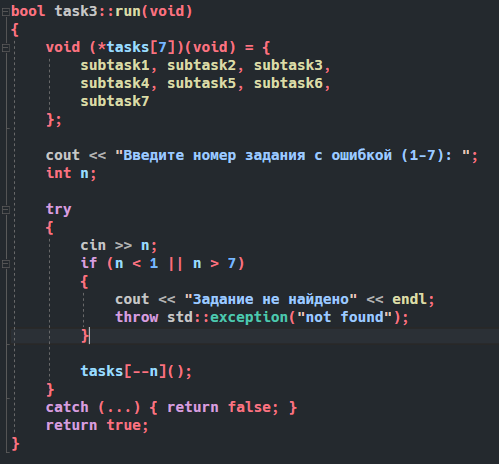


Рисунок 6 － Функция запуска подзадачи

1. Задача №1－Дано натурально число n. Получить все его натуральные делители.

Таблица 1 － обнаруженные ошибки в задаче №1

|  |  |  |
| --- | --- | --- |
| Список обнаруженных ошибок | | |
| Код | Ошибка | Метод исправления |
| С2065 | i: необъявленный идентификатор | Объявить переменную i целочисленного типа и установить ей значение равное 1 при инициализации цикла for |
| С2065 | n: необъявленный идентификатор | В теле функции main объявить пере |
| С4430 | Отсутствует идентификатор типа функции | Определить возвращаемый тип функции «main» |
| C2143 | Синтаксическая ошибка: отсутствие « ; » перед « ) » | В операторе for изменить знаки между секциями с « , » на « ; ». |
| C2146 | Синтаксическая ошибка: отсутствие « ; » перед идентификаторов div | Исправить условие остановки цикла условие на «i <= n». |
|  | Пустое тело цикла | После инициализации for заменить « ; » на составной оператор «{}» и поместить внутрь условную конструкцию для вывода делителей. |
| E0125 | Неправильное объявление условного выражения оператора if | Исправить выражение: оператор присваивания заменить на оператор сравнения; и обернуть выражение в круглые скобки. |

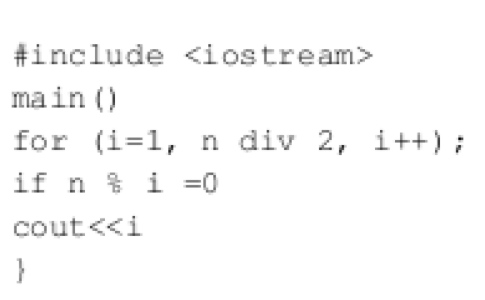


Рисунок 7 － Текст исходной программы задачи №1

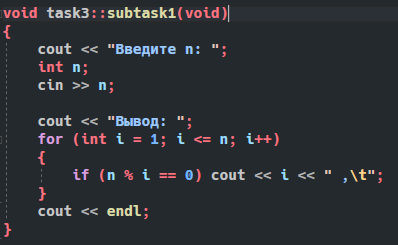


Рисунок 8 － Исправленный код задачи №1

1. Задача №2 － Вычислить величину «у», равную (n!!).

Таблица 2 － обнаруженные ошибки в задаче №2

|  |  |  |
| --- | --- | --- |
| Список обнаруженных ошибок | | |
| Код | Ошибка | Метод исправления |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

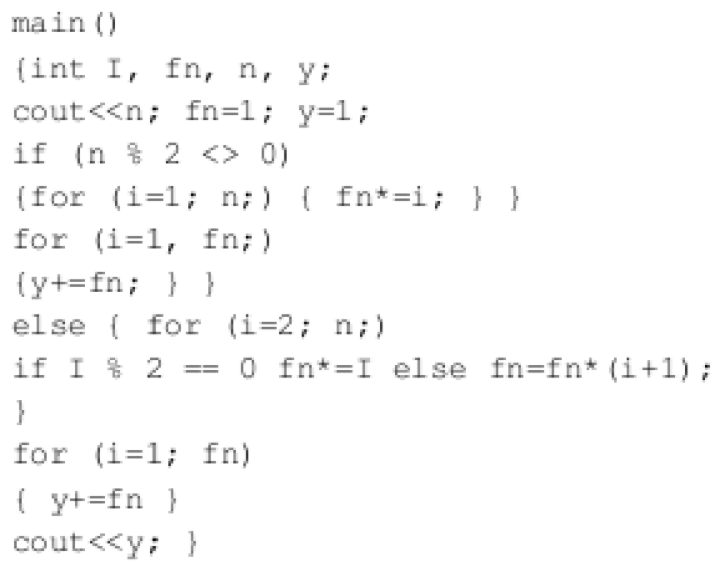


Рисунок 9 － Текст исходной программы задачи №2

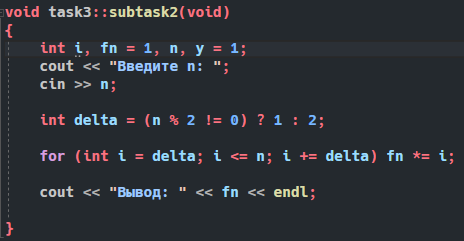


Рисунок 10 － Исправленный код задачи №2

1. Задача №3 － Дано натуральное число n. Вычислить произведение первых n сомножителей: .

Таблица 3 － обнаруженные ошибки в задаче №3

|  |  |  |
| --- | --- | --- |
| Список обнаруженных ошибок | | |
| Код | Ошибка | Метод исправления |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

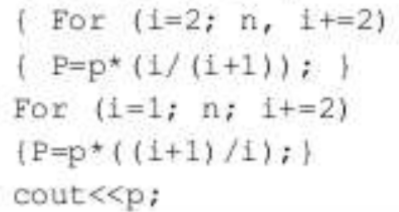


Рисунок 11 － Текст исходной программы задачи №3

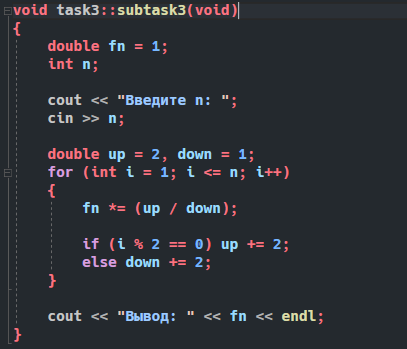


Рисунок 12 － Исправленный код задачи №3

1. Задача №4 － Вычислить .

Таблица 4 － обнаруженные ошибки в задаче №4

|  |  |  |
| --- | --- | --- |
| Список обнаруженных ошибок | | |
| Код | Ошибка | Метод исправления |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

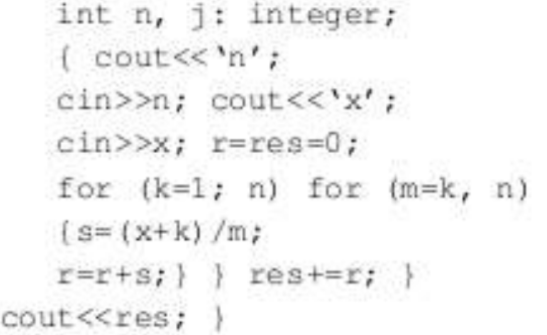


Рисунок 13 － Текст исходной программы задачи №4

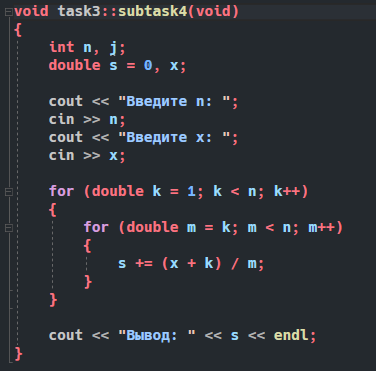


Рисунок 14 － Исправленный код задачи №4

1. Задача №5 － Вычислить .

Таблица 5 － обнаруженные ошибки в задаче №5

|  |  |  |
| --- | --- | --- |
| Список обнаруженных ошибок | | |
| Код | Ошибка | Метод исправления |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

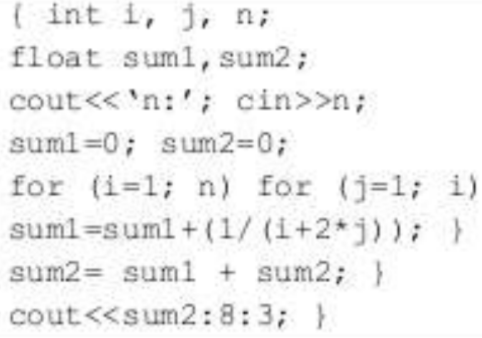


Рисунок 15 － Текст исходной программы задачи №5

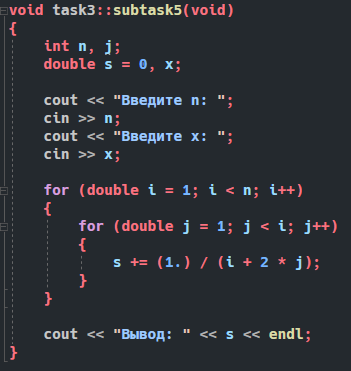


Рисунок 16 － Исправленный код задачи №5

1. Задача №6 － Дано 200 вещественных чисел. Определить, сколько из них больше своих «соседей», т. е предыдущего и последующего чисел.

Таблица 6 － обнаруженные ошибки в задаче №6

|  |  |  |
| --- | --- | --- |
| Список обнаруженных ошибок | | |
| Код | Ошибка | Метод исправления |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

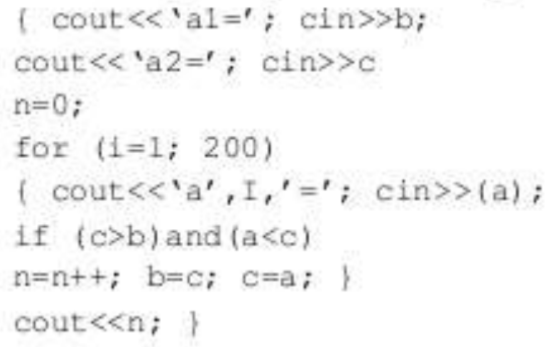


Рисунок 17 － Текст исходной программы задачи №6

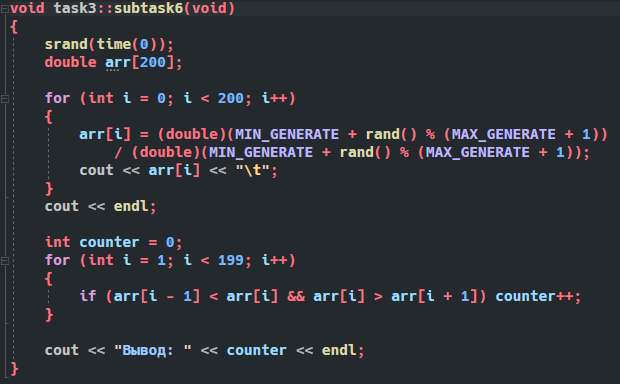


Рисунок 18 － Исправленный код задачи №6

1. Задача №7 －Вычислить .

Таблица 7 － обнаруженные ошибки в задаче №7

|  |  |  |
| --- | --- | --- |
| Список обнаруженных ошибок | | |
| Код | Ошибка | Метод исправления |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

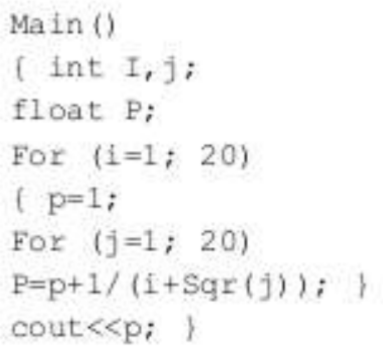


Рисунок 19 － Текст исходной программы задачи №7

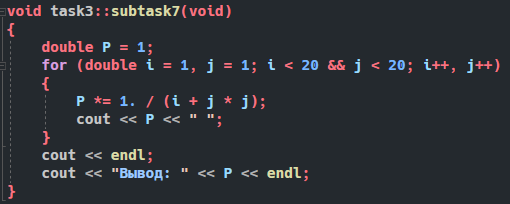


Рисунок 20 － Исправленный код задачи №7

* 1. Задание №4:

С начала необходимо объявить переменную вещественного типа и считать из потока ввода значение в неё. Далее через операторы условного ветвления «if-else» проверить на истинность выражение «переменная > 0» и в зависимости от результата выводить необходимое сообщение.

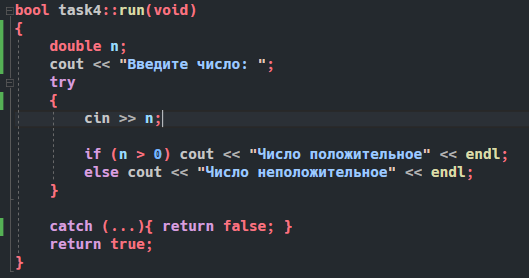


Рисунок 21 － Решение задания №4

* 1. Задание №5:

Для реализации меню определяется новая процедура «lab1::lab». Для начала необходимо определить переменные для хранения состояний: «exit\_trigger» － требования совершить выход из программы, «error\_handler» － обработчик ошибок.

Открывается бесконечный цикл, в теле которого производиться очистка при помощи функции «system» c аргументом «cls», инициализируется целочисленная переменная «task\_number» для определения номера требуемого задания и после в неё производиться считывание значения из потока ввода. После через оператор «switch» в зависимости от введённого значения выбирается соответствующее задание: производиться вызов функции «taskN::run()» (N － номер задания лабораторной работы) и возвращаемое значение записывается в переменную «error\_handler». Если значение равняется «5», то устанавливается значение истины для переменной «exit\_trigger»; если для значения не найден соответствующий «кейс», то на выводится необходимое сообщение.

После происходит проверка состояния для совершения выхода из цикла. Если выражение ложно, проверяется наличие ошибок при выполнении функций с заданиями: если переменная для обработки ошибок имеет истинное значение, то выводится сообщение, что во время выполнения задания возникло исключение (ошибка).

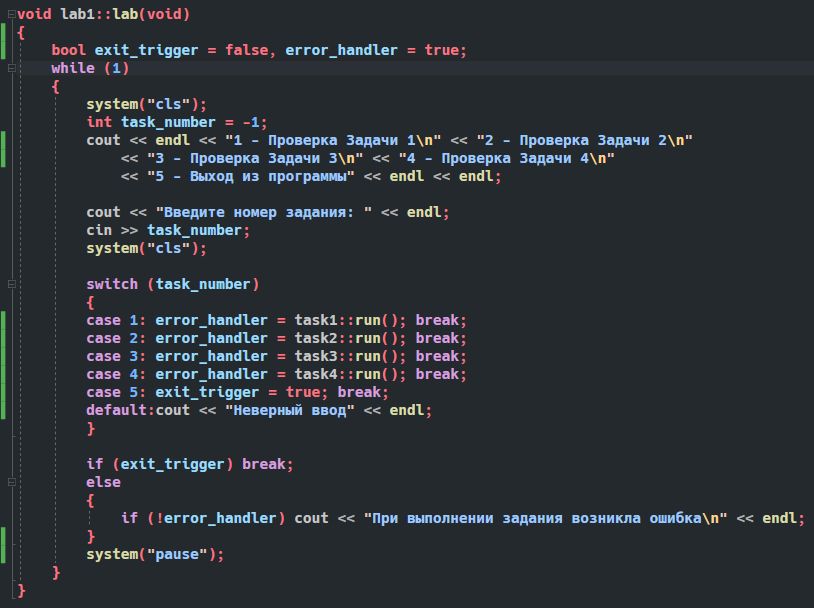


Рисунок 22 － Решение задания №5

Вывод: научились работать с базовыми конструкциями языка С++ и управляющими конструкциями.

Приложение А

Листинг программного кода языка C++

Файл «lab1.h»

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <locale>

#include <cmath>

// для задания 2

#define EBSILON 0.001

// для генерации чисел

#define MAX\_GENERATE 1000

#define MIN\_GENERATE 100

namespace lab1

{

using namespace std;

namespace task1

{

double function(double a, double b);

bool run(void);

}

namespace task2

{

double function(double x);

bool run(void);

}

namespace task3

{

void subtask1(void);

void subtask2(void);

void subtask3(void);

void subtask4(void);

void subtask5(void);

void subtask6(void);

void subtask7(void);

bool run(void);

}

namespace task4

{

bool run(void);

}

void lab(void);

}

Файл «lab1.cpp»

#include "lab1.h"

using namespace lab1;

double task1::function(double a, double b)

{

return atan((sin(pow(a, 2.)) + pow(b, 3.)) / sqrt(1. + (b) / (a + cos(M\_PI \* pow(b, 2.)))));

}

bool task1::run(void)

{

cout << fixed;

cout.precision(4);

double a, b;

cout << "Введите числа (a b): ";

try

{

cin >> a >> b;

cout << "Сумма чисел A и B равна " << function(a, b) << endl;

}

catch (...){ return false; }

return true;

}

double task2::function(double x)

{

double current\_state = 0, iter, sum = 0;

for (int n = 1; true; n++)

{

iter = pow(-1., n - 1) \* (pow(x, 2. \* n - 1) / (2. \* n - 1));

sum += iter;

if (current\_state != 0)

{

if (fabs(current\_state - iter) <= EBSILON) break;

}

current\_state = iter;

}

return sum;

}

bool task2::run(void)

{

double x;

cout << "Введите X (-1 <= x <= 1): ";

try

{

cin >> x;

if (fabs(x) > 1)

{

cout << "Значение не соответствует отрезку" << endl;

system("pause");

throw std::exception("value out of range");

}

cout << "F(x): " << function(x) << endl;

}

catch (...){ return false; }

return true;

}

void task3::subtask1(void)

{

cout << "Введите n: ";

int n;

cin >> n;

cout << "Вывод: ";

for (int i = 1; i <= n; i++)

{

if (n % i == 0) cout << i << " ,\t";

}

cout << endl;

}

void task3::subtask2(void)

{

int i, fn = 1, n, y = 1;

cout << "Введите n: ";

cin >> n;

int delta = (n % 2 != 0) ? 1 : 2;

for (int i = delta; i <= n; i += delta) fn \*= i;

cout << "Вывод: " << fn << endl;

}

void task3::subtask3(void)

{

double fn = 1;

int n;

cout << "Введите n: ";

cin >> n;

double up = 2, down = 1;

for (int i = 1; i <= n; i++)

{

fn \*= (up / down);

if (i % 2 == 0) up += 2;

else down += 2;

}

cout << "Вывод: " << fn << endl;

}

void task3::subtask4(void)

{

int n, j;

double s = 0, x;

cout << "Введите n: ";

cin >> n;

cout << "Введите x: ";

cin >> x;

for (double k = 1; k < n; k++)

{

for (double m = k; m < n; m++)

{

s += (x + k) / m;

}

}

cout << "Вывод: " << s << endl;

}

void task3::subtask5(void)

{

int n, j;

double s = 0, x;

cout << "Введите n: ";

cin >> n;

cout << "Введите x: ";

cin >> x;

for (double i = 1; i < n; i++)

{

for (double j = 1; j < i; j++)

{

s += (1.) / (i + 2 \* j);

}

}

cout << "Вывод: " << s << endl;

}

void task3::subtask6(void)

{

srand(time(0));

double arr[200];

for (int i = 0; i < 200; i++)

{

arr[i] = (double)(MIN\_GENERATE + rand() % (MAX\_GENERATE + 1))

/ (double)(MIN\_GENERATE + rand() % (MAX\_GENERATE + 1));

cout << arr[i] << "\t";

}

cout << endl;

int counter = 0;

for (int i = 1; i < 199; i++)

{

if (arr[i - 1] < arr[i] && arr[i] > arr[i + 1]) counter++;

}

cout << "Вывод: " << counter << endl;

}

void task3::subtask7(void)

{

double P = 1;

for (double i = 1, j = 1; i < 20 && j < 20; i++, j++)

{

P \*= 1. / (i + j \* j);

cout << P << " ";

}

cout << endl;

cout << "Вывод: " << P << endl;

}

bool task3::run(void)

{

void (\*tasks[7])(void) = {

subtask1, subtask2, subtask3,

subtask4, subtask5, subtask6,

subtask7

};

cout << "Введите номер задания с ошибкой (1-7): ";

int n;

try

{

cin >> n;

if (n < 1 || n > 7)

{

cout << "Задание не найдено" << endl;

throw std::exception("not found");

}

tasks[--n]();

}

catch (...) { return false; }

return true;

}

bool task4::run(void)

{

double n;

cout << "Введите число: ";

try

{

cin >> n;

if (n > 0) cout << "Число положительное" << endl;

else cout << "Число неположительное" << endl;

}

catch (...){ return false; }

return true;

}

void lab1::lab(void)

{

bool exit\_trigger = false, error\_handler = true;

while (1)

{

system("cls");

int task\_number = -1;

cout << endl << "1 - Проверка Задачи 1\n" << "2 - Проверка Задачи 2\n"

<< "3 - Проверка Задачи 3\n" << "4 - Проверка Задачи 4\n"

<< "5 - Выход из программы" << endl << endl;

cout << "Введите номер задания: " << endl;

cin >> task\_number;

system("cls");

switch (task\_number)

{

case 1: error\_handler = task1::run(); break;

case 2: error\_handler = task2::run(); break;

case 3: error\_handler = task3::run(); break;

case 4: error\_handler = task4::run(); break;

case 5: exit\_trigger = true; break;

default:cout << "Неверный ввод" << endl;

}

if (exit\_trigger) break;

else

{

if (!error\_handler) cout << "При выполнении задания возникла ошибка\n" << endl;

}

system("pause");

}

}

Файл «main.cpp»

#include <iostream>

#include <locale>

#include "labs/lab1/lab1.h"

int main(void)

{

setlocale(LC\_ALL, "rus");

lab1::lab();

return 0;

}