

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

Дисциплина: «Программирование»

Отчёт по лабораторной работе № 4

Лабораторная работа №4. Использование библиотек динамической компоновки.

**Выполнил студент группы №485:**

**Проверили:**

Корниенко Иван Григорьевич

Федин Алексей Константинович

Санкт-Петербург

2019

# Постановка задачи

Необходимо написать программу, в которой для вычисления функции, используется динамически подключенная библиотека dll. В результате работы на экране пользователя должна быть выведена таблица с вычисленными значениями рядной () и стандартной () функций и их невязки () в интервале x*start* – x*end* для заданной точности ε. Вычисление ряда проводить до выполнения следующего условия: . Также нужно вывести значения невязки для x*ideal* со значениями точности ε от 0.1 до 0.17.

# Исходные данные

В качестве исходных данных используются начальный x*start*, шаг ∆x, конечный x*end*, точность ε, x*ideal*, а также параметр a, вводимые пользователем.

# Особые ситуации

* Если пользователь, при вводе числа, введёт вместо цифры любой другой символ, программа попросит ввести число заново.
* Если программа не найдёт dll библиотеку или одну из функций, будет выведена ошибка, и программа закроется.
* Так как задание предусматривает значение x от -1 до 1, то если пользователь при вводе любого x укажет значение меньше -1 или больше 1, то программа попросит ввести значение ещё раз.

# Математические методы и алгоритмы решения задач

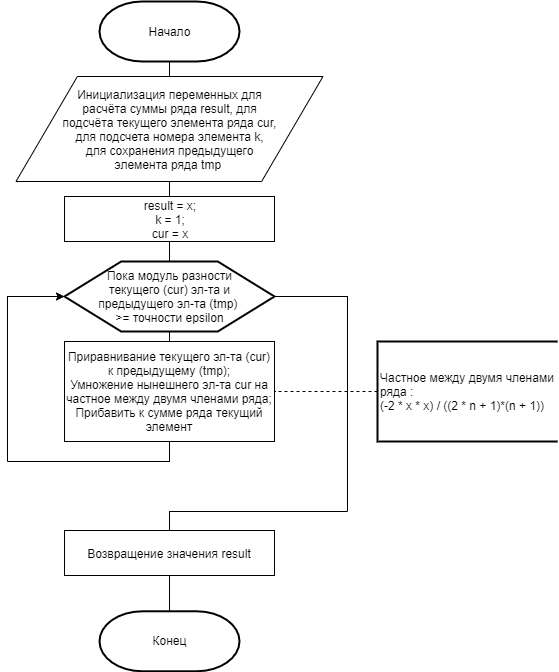


Рисунок 1 – Функция для расчёта суммы ряда

Согласно постановке задачи, для составления программы будут использоваться алгоритмы, блок-схема которой представлена выше.

Программа использует следующие переменные:

Таблица 1 – Переменные, используемы в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| running | bool | Определяет запущена ли программа |
| start | double | Начальная граница промежутка вычисления |
| step | double | Шаг |
| end | double | Конечная граница промежутка вычисления |
| ideal | double | Для расчёта невязки при eps от 0,1 до 0,1^7 |
| cur | long double | Вычисление текущего эл-та ряда |
| tmp | long double | Запоминание предыдущего эл-та ряда |
| result | long double | Сумма ряда |
| funcName | char\* | Имя функции |
| userInput | typename T | Проверка и возврат введённых пользователем данных |

# Структура программы

Из-за необходимости подключения библиотеки dll, решение было разделено на 2 проекта, один для создания dll, второй для самой программы. В проекте, отведённом для dll библиотеки (MathDll), содержится один модуль, отвечающий за расчёт функции (), суммы ряда, а также вывод имени функции. В проекте самой программы (Lab4) содержится 3 модуля. Один отвечает за запуск программы, вывод меню; второй за запрос значения от пользователя, взаимодействие с функциями dll; третий за проверку вводимых пользователем данных, изменение цвета текста консоли.

Модуль dllmain:

Таблица 2 – Функции, составляющие модуль dllmain проекта MathDLL

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| Полюшкина\_func | Расчёт суммы ряда |
| myf\_485 | Получение значения x, точности eps, вызов функции, производящей расчёт суммы ряда |
| myf\_math | Расчёт путём функций библиотеки cmath |
| FName | Вывод имени функции |

Модуль Source:

Таблица 3 – Функции, составляющие модуль Source проекта Lab4

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| main | Запуск программы |

Модуль Calculation:

Таблица 4 – Функции, составляющие модуль Calculation проекта Lab4

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| Calculation | Выполнение задания и вывод результатов в консоль |

Модуль VarCheckAndColorSet:

Таблица 5 – Функции, составляющие модуль VarCheckAndColorSet проекта Lab4

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| SetColor | Изменение цвета текста в консоли |
| CoutWithColor | Вывод сообщения с заданным цветом |
| GetInput | Проверка и возврат введённых пользователем данных |
| GetBool | Ввод пользователем значения типа bool |
| GetDouble | Ввод пользователем значения типа double |
| GetInt | Ввод пользователем значения типа int |

# Описание хода выполнения лабораторной работы

* В ходе лабораторной работы было создано решение (Lab4) в интегрированной среде разработки Microsoft Visual Studio C++ 2019. В нём было создано 2 проекта.
* При получении пользовательского ввода необходимо было добавить проверку на его соответствие предполагаемому типу данных и условиям выбора.
* Перед повторением программы необходимо выгружать библиотеку dll для того, чтобы не возникало непредвиденных ошибок.

# Результат работы программы

В результате работы на экране пользователя должна быть выведена таблица с вычисленными значениями рядной и стандартной функций и их невязки в интервале x*start* – x*end* для заданной точности ε. Также будут выведены значения невязки для x*ideal* со значениями точности ε от 0.1 до 0.17.

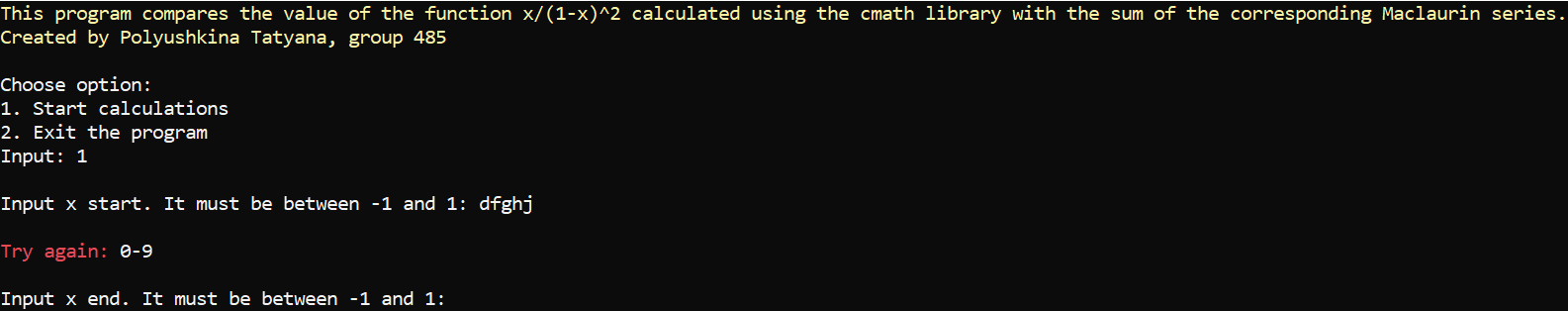


Рисунок 3 – Проверка вводимых пользователем данных

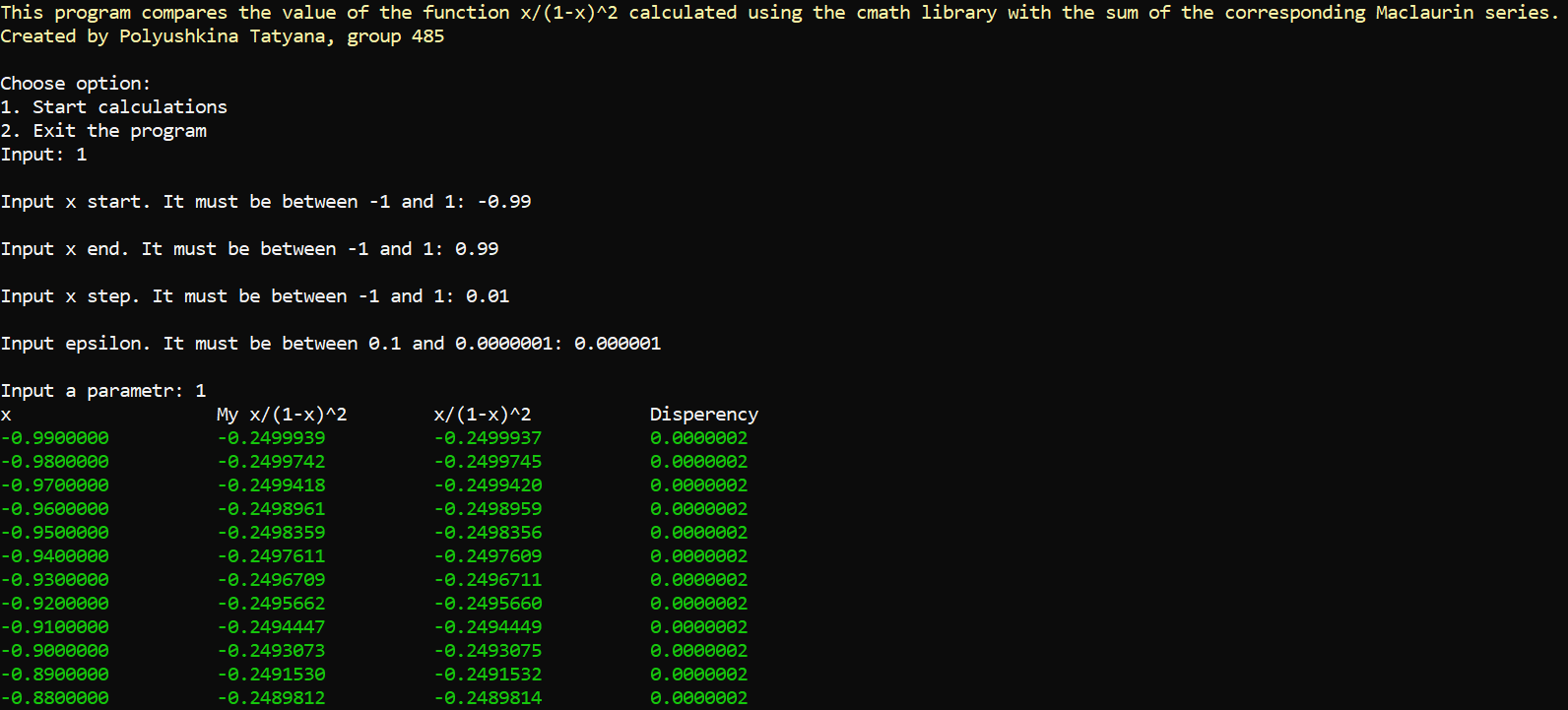


Рисунок 4 – Вывод результатов вычислений функций и их невязки для заданных границ x ( -1, 1), шага x (0.01) и точности (0,000001)

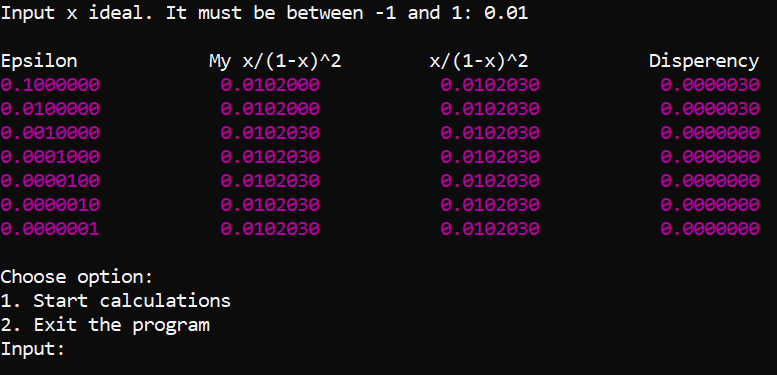


Рисунок 5 – Вывод результатов вычислений функций и их невязки для x-ideal (0,01).

# Текст программы

# [--- Начало программы ---]

**// dllmain.cpp**

**// Лабораторная работа №4.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#include "dllmain.h"

#include <cmath>

#include <string>

using namespace std;

//функция для вычисления рядной функции

double Полюшкина\_func(double x, double eps) {

long double result = x;

int k = 1;

long double cur = result;

long double tmp;

do {

tmp = cur;

cur \*= (x \* k + x) / k;

result += cur;

++k;

} while (abs(cur - tmp) >= eps);

return result;

}

// для вызова внутренней функции

double \_\_declspec(dllexport) myf\_485(double x, double a, double eps) {

return Полюшкина\_func(x, eps);

}

//вычисление показательной функции

double \_\_declspec(dllexport) myf\_math(double x, double a) {

return x / pow(1 - x, 2);

}

//возвращает имя функции

char \_\_declspec(dllexport)\* FName() {

char tmp[] = "x/(1-x)^2\0";

char\* funcName = new char[11];

for (int i = 0; i < 11; ++i)

funcName[i] = tmp[i];

return funcName;

}

**// dllmain.h**

**// Лабораторная работа №4.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

extern "C" {

//вычисление рядной функции

double \_\_declspec(dllexport) myf\_485(double x, double a, double eps);

//вычисление показательной функции

double \_\_declspec(dllexport) myf\_math(double x, double a);

//возвращает имя функции

char \_\_declspec(dllexport)\* FName();

}

**// Source.cpp**

**// Лабораторная работа №4.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#include <iostream>

#include "VarCheckAndColorSet.h"

#include "Calculation.h"

using namespace std;

enum MenuItems {

minMenuItem = 0, keepRunning, close, maxMenuItem

};

bool Menu() { // Вывод меню и проверка введённых данных

cout << "Choose option:\n" << "1. Start calculations\n" << "2. Exit the program\n" << "Input: ";

int menuItem; // пункт меню для ввода пользователем

bool running = true; // Для возврата в main, проверка запущена ли программа

while (true) { // ввод пользователем элемента меню и его проверка

menuItem = GetInt(); // ввод элемента меню, проверка на int

if (menuItem > minMenuItem && menuItem < maxMenuItem) { // проверка на выход за пределы меню

break;

}

else CoutWithColor(red, "Try again: ");

}

if (menuItem == keepRunning) {

running = Calculation();

}

else

running = false;

return running; // если running = true - программа продолжается, если false, то программа завершается

}

int main() { // работа программы

CoutWithColor(yellow, "This program compares the value of the function x/(1-x)^2 calculated using the cmath library with the sum of the corresponding Maclaurin series.\nCreated by Polyushkina Tatyana, group 485\n\n");

bool running = true; // Для определения запущенности программы

while (running)

{

running = Menu();

}

return 0;

}

**// Calculation.cpp**

**// Лабораторная работа №4.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include "VarCheckAndColorSet.h"

using namespace std;

typedef double(\*Myf\_485)(double x, double a, double eps); //вычисление рядной функции

typedef double(\*Myf\_math)(double x, double a); //вычисление функции из math.h

typedef char\* (\*FName)(); //запрос имени функции

bool Calculation() {

bool running = true;

HINSTANCE hGetProcIDDLL = LoadLibrary(TEXT("MathDLL.dll")); //загрузка библиотеки

if (!hGetProcIDDLL) {

cout << "Error opening dll. Press any key to exit." << endl;

running = false;

return running;

}

Myf\_485 myf\_485 = (Myf\_485)GetProcAddress(hGetProcIDDLL, "myf\_485"); //загрузка функции

if (!myf\_485) {

cout << "Error loading function. Press any key to exit." << endl;

running = false;

return running;

}

FName Fname = (FName)GetProcAddress(hGetProcIDDLL, "FName"); //загрузка функции

if (!Fname) {

cout << "\nError loading function. Press any key to exit." << endl;

running = false;

return running;

}

Myf\_math myf\_math = (Myf\_math)GetProcAddress(hGetProcIDDLL, "myf\_math"); //загрузка функции

if (!myf\_math) {

cout << "\nError loading function. Press any key to exit." << endl;

running = false;

return running;

}

double start, step, end, eps, a;

do {

cout << "\nInput x start. It must be between -1 and 1: ";

start = GetDouble();

} while (start >= 1 || start <= -1);

do {

cout << "\nInput x end. It must be between -1 and 1: ";

end = GetDouble();

} while (end < start || end >= 1 || end <= -1);

do {

cout << "\nInput x step. It must be between -1 and 1: ";

step = GetDouble();

} while (start + step <= start || step >= 1 || step <= -1);

do {

cout << "\nInput epsilon. It must be between 0.1 and 0.0000001: ";

eps = GetDouble();

} while (eps > 0.1 || eps < 10e-7);

cout << "\nInput a parametr: ";

a = GetDouble();

int startInt = start \* 10e7;

int stepInt = step \* 10e7;

int endInt = end \* 10e7;

char\* funcName = Fname();

cout << setw(20) << left << "x" << setw(3) << left << "My " << setw(17) << left << funcName << setw(20) << left << funcName << setw(20) << left << "Disperency" << endl;

SetColor(green);

cout << fixed << setprecision(7);

double x = start;

double myf\_485Result, myf\_mathResult, disperency;

for (int i = startInt; i <= endInt; i += stepInt, x += step) {

myf\_485Result = myf\_485(x, a, eps);

myf\_mathResult = myf\_math(x, a);

disperency = abs(myf\_485Result - myf\_mathResult);

if (disperency > 1) {

cout << setw(20) << left << x << setw(20) << left << 0 << setw(20) << left << myf\_mathResult << setw(20) << left << 0 << endl;

}

else

cout << setw(20) << left << x << setw(20) << left << myf\_485Result << setw(20) << left << myf\_mathResult << setw(20) << left << disperency << endl;

}

SetColor(white);

double ideal;

do {

cout << "\nInput x ideal. It must be between -1 and 1: ";

ideal = GetDouble();

} while (ideal >= 1 || ideal <= -1);

eps = 0.1;

cout << setw(20) << left << "\nEpsilon" << setw(3) << left << "My " << setw(17) << left << funcName << setw(20) << left << funcName << setw(20) << left << "Disperency" << endl;

SetColor(purple);

for (int i = 0; i < 7; ++i, eps /= 10) {

myf\_485Result = myf\_485(ideal, a, eps);

myf\_mathResult = myf\_math(ideal, a);

cout << setw(20) << left << eps << setw(20) << left << myf\_485Result << setw(20) << left << myf\_mathResult << setw(20) << left << abs(myf\_485Result - myf\_mathResult) << endl;

}

SetColor(white);

cout << endl;

delete[] funcName;

FreeLibrary(hGetProcIDDLL);

return running;

}

**// VarCheckAndColorSet.cpp**

**// Лабораторная работа №4.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#include <iostream>

#include <Windows.h>

#include <string>

#include "VarCheckAndColorSet.h"

using namespace std;

template <typename T> // создание шаблона T

T InputCheck() { // проверка ввода пользователя

T userInput; // создание переменной заданного типа для записи ввода пользователя

cin >> userInput;

while (cin.fail()) { // пока введенные данные не соответствуют типу

CoutWithColor(red, "\nTry again: ");

cin.clear(); // очищает cin.fail()

cin.ignore(INT\_MAX, '\n'); // очистка буфера

cin >> userInput;

}

cin.ignore(INT\_MAX, '\n');

return userInput;

}

int GetInt() { // ввод пользователем, проверка и возврат значения int

return InputCheck<int>();

}

bool GetBool() { // ввод пользователем, проверка и возврат значения bool

return InputCheck<bool>();

}

double GetDouble() {

return InputCheck<double>();

}

void SetColor(const int& color) { // изменение цвета в консоли

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color);

}

void CoutWithColor(const int& color, const string& message) { // вывод сообщения message с цветом color

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color); // изменение цвета на color

cout << message; // вывод сообщения

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), white); // изменение цвета на белый

}

**// Calculation.h**

**// Лабораторная работа №4.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#pragma once

bool Calculation();

**// VarCheckAndColorSet.h**

**// Лабораторная работа №4.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#pragma once

#include <string>

using namespace std;

enum Color { blue = 9, green, azure, red, purple, yellow, white };

int GetInt();

bool GetBool();

double GetDouble();

void CoutWithColor(const int& color, const string& message);

void SetColor(const int& color);

# [--- Конец программы ---]