Министерство науки и высшего образования РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа киберфизических систем и управления

УДК 004.421

УТВЕРЖДАЮ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_г.

Отчет

по дисциплине «Теория и технология программирования»

Лабораторная работа №9

«Использование библиотек динамической компоновки», вариант 17

Выполнил:

Студент

гр.3530902/90001 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Непушкин С.А. подпись, дата

Доцент ВШКФСиУ,

Кандидат технических наук

С. В. Хлопин \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Санкт-Петербург

2020

**1. Цель работы.**

Цель задания – Ознакомиться с возможностью и методами использования библиотек динамической компоновки dll (Dynamic Linked Library).

**2. Задание:**

Написать программу в которой для вычисления функции, используется динамически подключенная библиотека dll. Вычисление ряда проводить до условия минимизации значения разности двух соседних членов ряда меньше заданного эпсилон. Вычислить значения невязки значений рядной и стандартной Windows ("math.h") функций , проанализировать динамику изменения значения невязки в зависимости от количества слагаемых в ряде. Для остановки счета рядов необходимо использовать следующее условие:.

**3. Текст программы**

Проект DllTester

Source.cpp:

#include<Windows.h>

#include<iostream>

#include<locale.h>

#include<exception>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

cin.exceptions(cin.failbit | cin.badbit);

HINSTANCE dll\_link = LoadLibrary("Nepushkin.dll");

typedef char\*(\*FuncName)();

FuncName name = (FuncName)GetProcAddress(dll\_link, "FName");

typedef double(\*TrigFunction)(double d1, double d2, double epsilon);

TrigFunction nepushkin\_tg = (TrigFunction)GetProcAddress(dll\_link, name());

typedef double(\*TrigMathFunction)(double d1, double d2);

TrigMathFunction nepushkin\_math\_tg = (TrigMathFunction)GetProcAddress(dll\_link, "myf\_math");

double xstart;

double xend;

double xstep;

double xideal;

double eps;

try {

cout << "Введите x\_start = ";

cin >> xstart;

cout << "Введите x\_end = ";

cin >> xend;

cout << "Введите x\_step = ";

cin >> xstep;

cout << "Введите epsilon = ";

cin >> eps;

}

catch (exception e)

{

cout << "Ошибка ввода данных";

return 0;

}

printf("\n");

printf(" x | myFunction(x) | Function(x) | delta \n");

double x;

for (x = xstart; x < xend - xstep / 10; x = x + xstep)

{

double func = nepushkin\_math\_tg(x, 0);

double myfunc = nepushkin\_tg(x, 0, eps);

double delta = (func - myfunc)\*(func + myfunc);

if (delta < 0) delta \*= -1;

delta = sqrt(delta);

printf("%9.2lf | %28.10lf | %28.10lf | %10.20lf\n", x, myfunc, func, delta);

}

double func = nepushkin\_math\_tg(xend, 0);

double myfunc = nepushkin\_tg(xend, 0, eps);

double delta = (func - myfunc)\*(func + myfunc);

if (delta < 0) delta \*= -1;

delta = sqrt(delta);

printf("%9.2lf | %28.10lf | %28.10lf | %10.20lf\n", xend, myfunc, func, delta);

try {

cout << "Введите x\_ideal = ";

cin >> xideal;

}

catch (exception e)

{

cout << "Ошибка ввода данных";

return 0;

}

eps = 1;

printf("\n");

printf(" epsilon | myFunction(x) | Function(x) | delta \n");

for (int i = 0; i < 7; i++)

{

eps /= 10;

double func = nepushkin\_math\_tg(xideal, 0);

double myfunc = nepushkin\_tg(xideal, 0, eps);

double delta = (func - myfunc)\*(func + myfunc);

if (delta < 0) delta \*= -1;

delta = sqrt(delta);

printf("%9.7lf | %28.10lf | %28.10lf | %10.20lf\n", eps, myfunc, func, delta);

}

return 0;

}

Проект Nepushkin

dllmain.cpp:

#include "stdafx.h"

BOOL APIENTRY DllMain( HMODULE hModule,

DWORD ul\_reason\_for\_call,

LPVOID lpReserved

)

{

switch (ul\_reason\_for\_call)

{

case DLL\_PROCESS\_ATTACH:

case DLL\_THREAD\_ATTACH:

case DLL\_THREAD\_DETACH:

case DLL\_PROCESS\_DETACH:

break;

}

return TRUE;

}

Nepushkin.cpp:

#include "stdafx.h"

#include "Nepushkin.h"

#define PI 3.14159265358979323846

double bernulli[50];

int bernulliCount = 0;

void calcBernulli(int n)

{

double res = 0;

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

{

int pow = ((i - 1) % 2 == 1 ? -1 : 1);

int numerator = 1;

for (int j = 2 \* n - 2 \* i + 2; j <= 2 \* n; j++)

{

numerator \*= j;

}

int denominator = 1;

for (int j = 1; j <= 2 \* i; j++)

{

denominator \*= j;

}

res += pow \* bernulli[i] \* numerator / denominator;

}

int pow = (n % 2 == 1 ? -1 : 1);

bernulli[n] = pow \* ((double)1 / (2 \* n + 1) - (double)1 / 2 + res);

}

double Bernulli(int n)

{

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

{

calcBernulli(i);

}

return bernulli[n];

}

double nepushkin\_func(double x, double epsilon)

{

double result = 0;

int pow = 4;

int n = 1;

double delta = 0;

double multfact = x / 2;

do

{

delta = abs(Bernulli(n)) \* pow \* (pow - 1) \* multfact;

result += delta;

pow = pow << 2;

n++;

multfact \*= x \* x / (2 \* n \* (2 \* n - 1));

} while (abs(delta) >= epsilon);

return result;

}

double myf\_math(double x, double y)

{

return tan(x);

}

double myf\_90001(double x, double y, double epsilon)

{

while (x > PI / 2) x -= PI;

while (x < -PI / 2) x += PI;

if (abs(x) > PI / 4)

{

return 1 / myf\_90001(PI / 2 - x, y, epsilon);

}

if (abs(x) <= epsilon / 10) return x;

if (x < 0) return -nepushkin\_func(-x, epsilon);

return nepushkin\_func(x, epsilon);

}

char\* FName()

{

return (char\*) "myf\_90001";

}

stdafx.cpp:

#include "stdafx.h"

Nepushkin.h:

#pragma once

extern "C" \_\_declspec(dllexport)

double myf\_math(double x, double y);

extern "C" \_\_declspec(dllexport)

double myf\_90001(double x, double y, double epsilon);

extern "C" \_\_declspec(dllexport)

char\* FName();

stdafx.h:

// stdafx.h: включаемый файл для стандартных системных включаемых файлов

// или включаемых файлов для конкретного проекта, которые часто используются, но

// нечасто изменяются

//

#pragma once

#include "targetver.h"

#define WIN32\_LEAN\_AND\_MEAN // Исключите редко используемые компоненты из заголовков Windows

// Файлы заголовков Windows

#include <windows.h>

// установите здесь ссылки на дополнительные заголовки, требующиеся для программы

#include<cmath>

targetver.h:

#pragma once

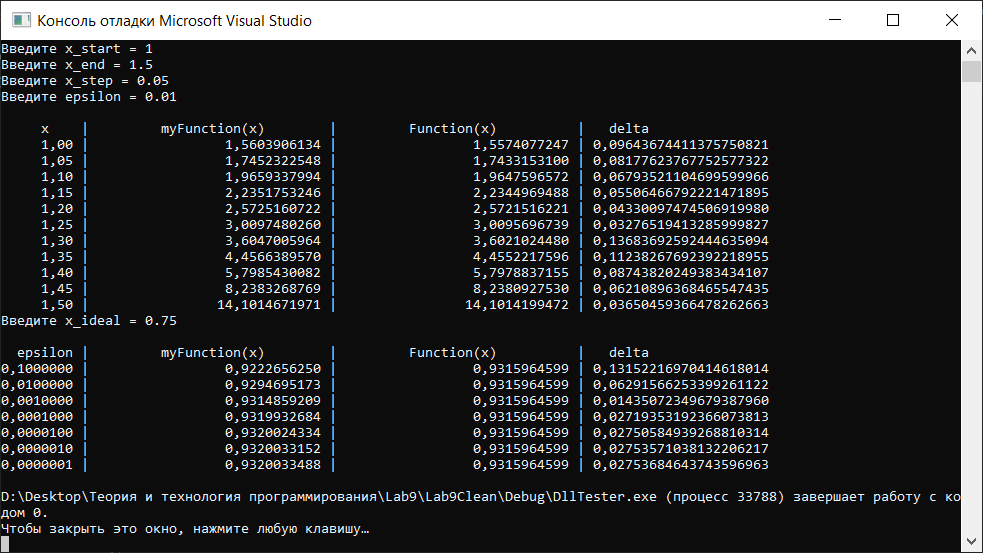
// Включение SDKDDKVer.h обеспечивает определение самой последней доступной платформы Windows.

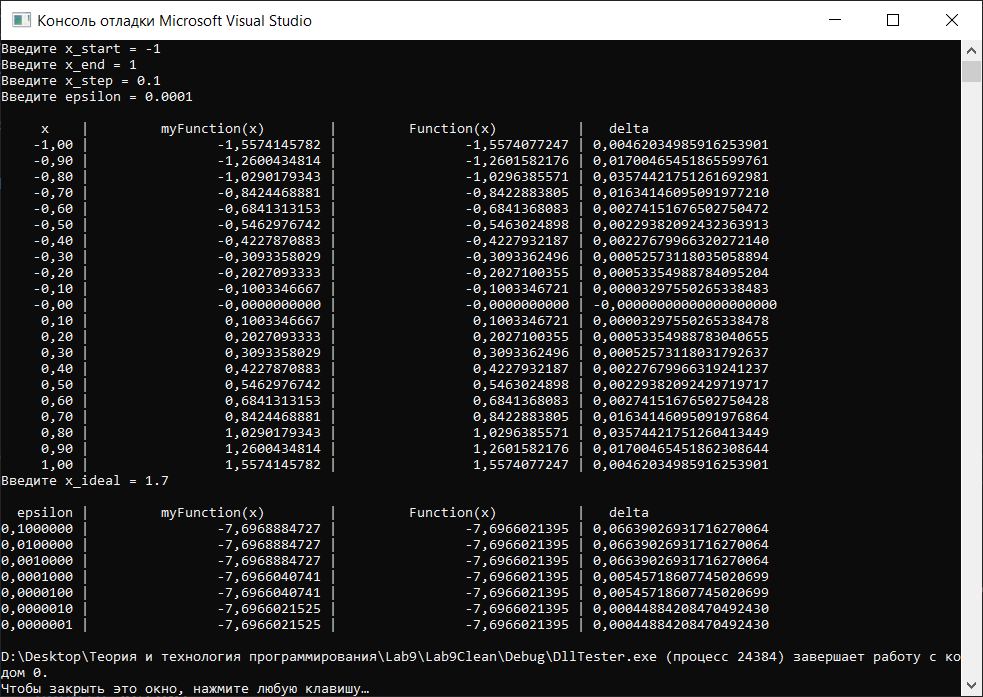
// Если требуется выполнить сборку приложения для предыдущей версии Windows, включите WinSDKVer.h и

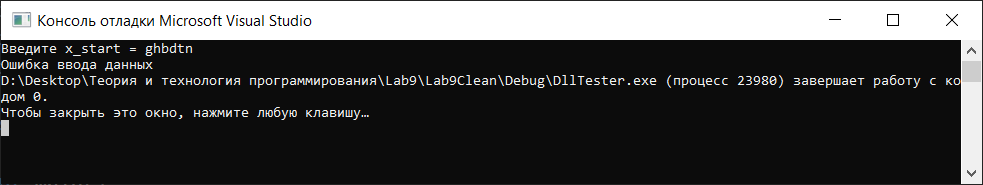
// задайте для макроса \_WIN32\_WINNT значение поддерживаемой платформы перед включением SDKDDKVer.h.

#include <SDKDDKVer.h>

**4. Пример работы программы**







**Вывод**

В ходе лабораторной работы я:

-научился считать число Бернулли

-столкнулся с проблемой вывода функции до заданной точности, но успешно решил ее

-узнал о типе исключений “exceptin”

-узнал для чего используются файлы заголовков

- столкнулся с проблемой обнуления переменной при решении квадратного уравнения, но успешно решил ее