

Федеральное государственное автономное образовательное учреждение высшего образования
«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 технологии программирования»

Лабораторная работа №9

Выполнил:

студент гр. 3532703/00001

_____ Д.П. Кимельфельд
подпись, дата

Санкт-Петербург 2021 г.

Оглавление

Введение	3
1 Описание программы.....	4
2 Текст программы	5
2.1 Текст основной программы	5
2.2 Текст динамической библиотеки DLL	7
3 Пример работы программы.....	8
Вывод.....	Ошибка! Закладка не определена.

Введение

1. Цель работы

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

2. Задание

Написать программу для вычисления значения функции, заданной с помощью ряда. Вычисление ряда проводить до условия минимизации значения разности двух соседних членов ряда, меньше заданного эpsilon. Вычислить значения невязки значений рядной и стандартной Windows

("math.h") функций $\delta = \sqrt{|MyFunc(x)^2 - Function(x)^2|}$, проанализировать динамику изменения значения невязки в зависимости от количества слагаемых в ряде. Для остановки счета рядов необходимо использовать следующее условие: $:|F(x_n)| < eps$.

Вариант № 13.

Вычисление котангенса.

$$\operatorname{cosec}^2(x) = \frac{1}{x} + \sum_{n=0}^{\infty} \frac{2 * (2^{2n+1} - 1)}{(2 * n + 2)!}, (x^2 < \pi^2)$$

1 Описание программы

В рамках данной программы нужно было создать как саму программу, так и подключаемую к ней библиотеку DLL. Причем синтаксис функций, входящих в DLL, должен быть общим для всех участников нашей группы 3532703/00001, см. Приложение №1, так как каждая из динамических библиотек должна корректно работать с программой одногруппника. По своей сути DLL создана для эффективной организации памяти и дискового пространства, используя один экземпляр, то есть только один экземпляр, библиотечного модуля для различных приложений. В рамках нашей лабораторной работы мы лишь научились корректно создавать и подключать библиотеку динамической компоновки.

Пройдемся по пунктам программы.

1. Внутри библиотеки DLL есть 4 функции, 3 из которых внешние, то есть вызываются сторонними программами. Одна из функций отвечает за математическую операцию, лежащую в стандартной библиотеке. Вторая функция отвечает за счет ряда, и третья функция отвечает за имя функции. Имя функции передается в вызывающую программу с целью, чтобы человек понимал – ту ли библиотеку динамической компоновки он подключил.

2. В основной программе есть код только подключения библиотеки, проверки библиотеки на нужный формат, на кол-во функций.

3. В основной программе также лежит код проверок именно для моей функции, то есть, когда из DLL в основную программу поступает имя функции, я его принимаю и сравниваю с именем моей функции. Если они совпадают, то устанавливаю пользователю ограничения на мою функцию, если нет, то сообщая пользователю, что функция не моя, ограничений на неё установить не представляется возможным. Затем просто идет счет ряда функции и счет ряда с повышением точности для X_{ideal} и, конечно, вывод всего этого на консоль.

2 Текст программы

2.1 Текст основной программы

```
#include <iostream>
#include <iomanip>
#include <windows.h>
#include <math.h>

using namespace std;
// "Вариант №13. Вычисление косеканса в квадрате."
// Косеканс раскладывается в отрезках  $[-\pi+e, -e]$   $[e, \pi-e]$ , где в программе  $e = 1e-6$ 
// В программе число  $\pi = 3.141592$ 
void input(double &e, double &x0, double &x1, double &dx, double &a) {
    cout << "Введите точность эпсилон(E): ";
    cin >> e;
    cout << "Введите начальную координату отрезка: ";
    cin >> x0;
    cout << "Введите конечную координату отрезка: ";
    cin >> x1;
    cout << "Введите значение вашего шага (dX): ";
    cin >> dx;
    cout << "Введите значение параметра:";
    cin >> a;
}

double sigma(double x, double y) {
    double a = sqrt(fabs((x*x - y * y)));
    return a;
}

constexpr auto PI = 3.141592;
int main() // 1 - параметр, 2 - x, 3 - точность
{
    //-----//
    setlocale(LC_ALL, "Russian");
    string nameKimeIfeld = { "cosec(x)*cosec(x)" };
    typedef double(*myf_1A)(double, double, double);
    typedef double(*myf_math)(double, double);
    typedef const char* (*Fname)();
    //-----//
    string nameDll = "Dll1";
    if (!LoadLibraryExA(nameDll.c_str(), 0, LOAD_LIBRARY_AS_DATAFILE))
    {
        cout << "Ошибка в библиотеке" << endl;
        system("pause");
        return 0;
    }
    //
    HINSTANCE dll = LoadLibraryA("Dll1.dll");
    myf_1A cosec2 = (myf_1A)(GetProcAddress(dll, "myf_3532703_00001"));
    myf_math cosec2math = (myf_math)(GetProcAddress(dll, "myf_math"));
    Fname namevar = (Fname)(GetProcAddress(dll, "FName"));
    if (dll == NULL)
    {
        cout << "Ошибка библиотеки динамической компоновки!" << endl;
        system("pause");
        return 0;
    }
    if (!(cosec2 && cosec2math && namevar))
    {
        cout << "Ошибка в библиотеке" << endl;
        system("pause");
        return 0;
    }
    //-----//
    double epsilon, xStart, xFinish, dX, a;
```

```

string ss1 = namevar();
do {
    system("cls");

    input(epsilon, xStart, xFinish, dX,a);
} while ((xStart <= -1 * PI) || (xStart >= PI) || (xFinish <= -PI) || (xFinish >= PI) ||
(xStart > xFinish) || (epsilon <= 0.0) || (dX <= 0.0));
if (ss1 != nameKimelfeld) cout << "Подключена не моя библиотека...." << endl;
//-----//
long double x = xStart;
//-----//
cout << " -----"
--" << endl;
cout << " |" << setw(11) << "X" << "|" << "My " << setw(20) << namevar() << "|" <<
"math.h " << setw(21) << namevar() << "|" << setw(15) << "Sigma" << "|" << endl;
cout << " -----"
--" << endl;
while (x <= xFinish) {
    double myFun = cosec2(a, epsilon, x);
    double mathFun = cosec2math(a, x);
    if ((isnan(myFun)) || (isnan(mathFun)) || isinf(myFun) || isinf(mathFun)) {
        cout << " |" << setw(11) << x << "|" << setw(23) << "ERROR" << "|" <<
setw(28) << "ERROR" << "|" << setw(15) << "ERROR" << "|" << endl;
        cout << fixed;
        cout.precision(7);
        cout << " -----"
-----" << endl;
        x = round((x + dX) * 10000000) / 10000000;
        continue;
    }
    cout << fixed;
    cout << setprecision(7);
    cout << " |" << setw(11) << x << "|" << setw(23) << myFun << "|" << setw(28) <<
mathFun << "|" << setw(15) << sigma(myFun, mathFun) << "|" << endl;
    cout << " -----"
-----" << endl;
    x = round((x + dX) * 10000000) / 10000000;
}

//-----//
cout << endl << "Введите идеальный x: ";
cin >> x;
//-----//
cout << " -----"
--" << endl;
cout << " |" << setw(11) << "E" << "|" << "My " << setw(20) << namevar() << "|" << "math.h
" << setw(21) << namevar() << "|" << setw(15) << "Sigma" << "|" << endl;
cout << " -----"
--" << endl;

for (double j = 0.1; j > 0.0000001; j /= 10) {
    double myFun = cosec2(a, j, x);
    double mathFun = cosec2math(a, x);
    if ((isnan(myFun)) || (isnan(mathFun)) || isinf(myFun) || isinf(mathFun) ||
abs(x) >= PI ) {
        cout.precision(9);
        cout << " |" << setw(11) << x << "|" << setw(23) << "ERROR" << "|" <<
setw(28) << "ERROR" << "|" << setw(15) << "ERROR" << "|" << endl;
        cout << " -----"
-----" << endl;
        continue;
    }
    cout << fixed;
    cout << setprecision(7);

```

```

        cout << " |" << setw(11) << j << "|" << setw(23) << myFun << "|" << setw(28)
<< mathFun << "|" << setw(15) << sigma(myFun, mathFun) << "|" << endl;
        cout << " -----" << endl;
    }
    //-----//
    return 0;
}

```

2.2 Текст динамической библиотеки DLL

```

// dllmain.cpp : Определяет точку входа для приложения DLL.
#include "pch.h"
#include "string"
#include <vector>
#include <math.h>

extern "C" __declspec(dllexport) double myf_math(double a, double x) {
    return (1/sin(x))*(1/sin(x));
}

extern "C" __declspec(dllexport) const char * FName() {
    return "cosec(x)*cosec(x)";
}

extern "C" __declspec(dllexport) double Kimelfeld_cosec2(double a, double eps, double x) {
    double s = x, d = x;
    for (int n = 3; fabs(d) >= eps; n += 2){
        d *= -x * x / n / (n - 1);
        s += d;
    }
    return (1.0/s)*(1.0/s);
}
extern "C" __declspec(dllexport) double myf_3532703_00001(double a, double epsilon, double x) {
    return Kimelfeld_cosec2(a , epsilon, x);
}

```

3 Пример работы программы

```

C:\Users\HUAWEI\source\repos\ТПП 9\Debug\ТПП 9.exe
Введите точность эпсилон(E): 0.001
Введите начальную координату отрезка: -2
Введите конечную координату отрезка: 2
Введите значение вашего шага (dX): 0.25
Введите значение параметра: 5
-----
|      X|My      cosec(x)*cosec(x)|math.h      cosec(x)*cosec(x)|      Sigma| |
|---|---|---|---|---|
| -2.000000|      1.2094539|      1.2094504|      0.0028821|
|-----|-----|-----|-----|-----|
| -1.750000|      1.0327899|      1.0328142|      0.0070866|
|-----|-----|-----|-----|-----|
| -1.500000|      1.0050246|      1.0050289|      0.0029414|
|-----|-----|-----|-----|-----|
| -1.250000|      1.1104530|      1.1104056|      0.0102573|
|-----|-----|-----|-----|-----|
| -1.000000|      1.4122921|      1.4122829|      0.0050884|
|-----|-----|-----|-----|-----|
| -0.750000|      2.1522450|      2.1522437|      0.0023655|
|-----|-----|-----|-----|-----|
| -0.500000|      4.3506573|      4.3506853|      0.0156189|
|-----|-----|-----|-----|-----|
| -0.250000|     16.3375401|     16.3375417|      0.0072260|
|-----|-----|-----|-----|-----|
|  0.000000|      ERROR|      ERROR|      ERROR|
|-----|-----|-----|-----|-----|
|  0.250000|     16.3375401|     16.3375417|      0.0072260|
|-----|-----|-----|-----|-----|
|  0.500000|      4.3506573|      4.3506853|      0.0156189|
|-----|-----|-----|-----|-----|
|  0.750000|      2.1522450|      2.1522437|      0.0023655|
|-----|-----|-----|-----|-----|
|  1.000000|      1.4122921|      1.4122829|      0.0050884|
|-----|-----|-----|-----|-----|
|  1.250000|      1.1104530|      1.1104056|      0.0102573|
|-----|-----|-----|-----|-----|
|  1.500000|      1.0050246|      1.0050289|      0.0029414|
|-----|-----|-----|-----|-----|
|  1.750000|      1.0327899|      1.0328142|      0.0070866|
|-----|-----|-----|-----|-----|
|  2.000000|      1.2094539|      1.2094504|      0.0028821|
|-----|-----|-----|-----|-----|
Введите идеальный x:

```

Рисунок 1 – пример №1 выполнения программы со своим вариантом

```

Введите идеальный x: 1.75
-----
|      E|My      cosec(x)*cosec(x)|math.h      cosec(x)*cosec(x)|      Sigma| |
|---|---|---|---|---|
|  0.100000|      1.0336810|      1.0328142|      0.0423218|
|-----|-----|-----|-----|-----|
|  0.010000|      1.0336810|      1.0328142|      0.0423218|
|-----|-----|-----|-----|-----|
|  0.001000|      1.0327899|      1.0328142|      0.0070866|
|-----|-----|-----|-----|-----|
|  0.000100|      1.0328147|      1.0328142|      0.0009954|
|-----|-----|-----|-----|-----|
|  0.000010|      1.0328142|      1.0328142|      0.0001204|
|-----|-----|-----|-----|-----|
|  0.000001|      1.0328142|      1.0328142|      0.0001204|
|-----|-----|-----|-----|-----|
|  0.000000|      1.0328142|      1.0328142|      0.0000128|
|-----|-----|-----|-----|-----|

```

Рисунок 2 - пример №1 выполнения программы со своим вариантом (продолжение)


```

C:\Users\HUAWEI\source\repos\ТПП 9\64\Debug\ТПП 9.exe
Введите точность эпсилон(E): 0.01
Введите начальную координату отрезка: -2
Введите конечную координату отрезка: 1
Введите значение вашего шага (dX): 0.2
Введите значение параметра:5
Подключена не моя библиотека....
-----
|      X|My      x * cos(3 * x)|math.h      x * cos(3 * x)|      Sigma|
-----
| -2.000000|      -1.9205603|      -1.9203406|      0.0290514|
-----
| -1.800000|      -1.1421380|      -1.1424472|      0.0265764|
-----
| -1.600000|      -0.1404292|      -0.1399984|      0.0109915|
-----
| -1.400000|      0.6869579|      0.6863651|      0.0285322|
-----
| -1.200000|      1.0761539|      1.0761101|      0.0097067|
-----
| -1.000000|      0.9899396|      0.9899925|      0.0102309|
-----
| -0.800000|      0.5899741|      0.5899150|      0.0083512|
-----
| -0.600000|      0.1362636|      0.1363213|      0.0039628|
-----
| -0.400000|      -0.1449011|      -0.1449431|      0.0034883|
-----
| -0.200000|      -0.1650800|      -0.1650671|      0.0020619|
-----
|  0.000000|      0.0000000|      0.0000000|      0.0000000|
-----
|  0.200000|      0.1650800|      0.1650671|      0.0020619|
-----
|  0.400000|      0.1449011|      0.1449431|      0.0034883|
-----
|  0.600000|      -0.1362636|      -0.1363213|      0.0039628|
-----
|  0.800000|      -0.5899741|      -0.5899150|      0.0083512|
-----
|  1.000000|      -0.9899396|      -0.9899925|      0.0102309|
-----
Введите идеальный x:

```

Рисунок 3 - пример №2 выполнения программы с вариантом одnogруппника

Введите идеальный x: 1.8

E My	x * cos(3 * x) math.h	x * cos(3 * x)	Sigma
0.1000000	1.1464237	1.1424472	0.0954030
0.0100000	1.1421380	1.1424472	0.0265764
0.0010000	1.1424669	1.1424472	0.0067105
0.0001000	1.1424461	1.1424472	0.0015483
0.0000100	1.1424472	1.1424472	0.0003289
0.0000010	1.1424472	1.1424472	0.0000648
0.0000001	1.1424472	1.1424472	0.0000648

Рисунок 4 - пример №2 выполнения программы с вариантом одnogруппника (продолжение)

Документация к библиотеке DLL группа 3532703/00001

Название	Тип возвращаемо го значения	Принимаемые аргументы	Описание работы функции
myf_math	double	double /*1 параметр*/, double /*2 параметр*/ Параметр 1- дополнительный параметр для совместимости с другими библиотеками в данной функции не используется. Параметр 2- принимаемый аргумент	Функция возвращает $\sqrt{1+x}$ при этом функция взята из библиотеки «cmath»
myf_3532703_000 01	double	double /*1 параметр*/, double /*2 параметр*/, double /*3 параметр */ Параметр 1- дополнительный параметр для совместимости с другими библиотеками, в	Функция возвращает значение считаемого ряда функции $\sqrt{1+x}$ с введенной погрешностью