МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Разработка оконного приложения

Отчет

Лабораторная работа №3 по дисциплине

«Технологии программирования»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М.Л./

Киров 2021

**Задание**

Используя только функции WinAPI, разработать многооконное приложение, позволяющее высчитывать интеграл, заданный пользователем, методами прямоугольника и трапеций. Структура: 1 родительское окно и 2 дочерних окна.

**Листинг программы**

#pragma comment(linker,"\"/manifestdependency:type='win32' \

name='Microsoft.Windows.Common-Controls' version='6.0.0.0' \

processorArchitecture='\*' publicKeyToken='6595b64144ccf1df' language='\*'\"")

#include <iostream>

#include <Windows.h>

#include <string>

#include <math.h>

using namespace std;

string aa, n, a, b;

double s, h, I;

TCHAR buf[200];//Массив под строку в 4-м Edit

INT razmer;

char buff[200];

char znak[20];//Массви под знаки в функциях

double integ[20]; //Массив под функции

double trapp[20];

int arg[20];//Массив под аргументы функций

int bb,bbb; // счётчик функций

int ii, iii, iiii; //счётчики для всего

string cc;

int kek;

string k, kk;

double first, second;

TCHAR res1[10];

bool okno2 = false;

TCHAR res2[10];

int a1, a2, a3;

float wwe;

bool fflag;

char vv;

//double y;

bool okno;

int del;

bool kyk;

static pair<bool, HWND> AddWindow(const wstring&& winClass, const wstring&& title, HWND hParentWnd, const WNDPROC callback)

{

UnregisterClass(winClass.c\_str(), GetModuleHandle(nullptr));

WNDCLASSEX wc{ sizeof(WNDCLASSEX) };

HWND hWindow{};

wc.cbClsExtra = 0;

wc.cbWndExtra = 0;

wc.hbrBackground = reinterpret\_cast<HBRUSH>(GetStockObject(WHITE\_BRUSH));

wc.hCursor = LoadCursor(nullptr, IDC\_ARROW);

wc.hIcon = LoadIcon(nullptr, IDI\_APPLICATION);

wc.hIconSm = LoadIcon(nullptr, IDI\_APPLICATION);

wc.lpfnWndProc = callback;

wc.lpszClassName = winClass.c\_str();

wc.style = CS\_VREDRAW | CS\_HREDRAW;

const auto create\_window = [&hWindow, &winClass, &title, &hParentWnd]() -> pair<bool, HWND> {

if (hWindow = CreateWindow(winClass.c\_str(), title.c\_str(), WS\_OVERLAPPEDWINDOW & ~WS\_THICKFRAME & ~WS\_MAXIMIZEBOX,

(GetSystemMetrics(SM\_CXSCREEN) - 300) / 2,

(GetSystemMetrics(SM\_CYSCREEN) - 300) / 2,

450, 300, hParentWnd, nullptr, nullptr, nullptr); !hWindow)

return { false, nullptr };

ShowWindow(hWindow, SW\_SHOWDEFAULT);

UpdateWindow(hWindow);

return { true, hWindow };

};

if (!RegisterClassEx(&wc))

return create\_window();

return create\_window();

}

void cifra(string check, HWND dd)

{

if (razmer==0 || aa[razmer - 1]=='-' || aa[razmer-1]=='+' || isdigit(aa[razmer-1])) {

razmer = razmer + 1;

aa += check;

del = 1;

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

{

buf[i] = aa[i];

}

SetWindowText(dd, buf);

GetWindowText(dd, buf, razmer);

}

else Beep(600, 800);

}

void pryamoygolnik(int b, int a, int n, int a4, int arg)

{

switch (a4)

{

case 1: {

double s = 1.0\*(sin(a) + sin(b)) / 2.0;

double h = 1.0 \* (b - a) / n;

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

{

s += 1.0\* sin(a+i\*h) \* 1.0;

}

integ[bb] = 1.0 \* h \* s \* 1.0 \* arg;// \* 1.0;

bb = bb + 1;

//integ[0] = 12;

//trapp[0] = 22;

}

break;

case 2: {

double s = 1.0 \* (cos(a) + cos(b)) / 2.0;

double h = 1.0 \* (b - a) / n;

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

{

s += 1.0 \* cos(a + i \* h) \* 1.0;

}

integ[bb] = 1.0 \* h \* s \* arg \* 1.0;

bb = bb + 1;

}

break;

case 3: {

double s = 1.0 \* (sin(a)/cos(a) + sin(b)/cos(b)) / 2.0;

double h = 1.0 \* (b - a) / n \* 1.0;

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

{

s += 1.0 \* sin(a + i \* h)/cos(a+i\*h) \* 1.0;

}

integ[bb] = 1.0 \* h \* s \* arg \* 1.0;

bb = bb + 1;

}

break;

case 4: {

double s = 1.0 \* (cos(a)/sin(a) + cos(b)/sin(b)) / 2.0;

double h = 1.0 \* (b - a) / n \* 1.0;

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

{

s += 1.0 \* cos(a + i \* h)/sin(a + i \* h) \* 1.0;

}

integ[bb] = 1.0 \* h \* s \* arg \* 1.0;

bb = bb + 1;

}

break;

case 5: {

double s = 1.0 \* (a + b) / 2.0;

double h = 1.0 \* (b - a) / n \* 1.0;

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

{

s += 1.0 \* a + i \* h \* 1.0;

}

integ[bb] = 1.0 \* h \* s \* arg \* 1.0;

bb = bb + 1;

}

break;

case 6: {

double s = 1.0 \* (a\*a + b\*b) / 2.0;

double h = 1.0 \* (b - a) / n \* 1.0;

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

{

s += 1.0 \* (a + i \* h)\*(a+i\*h) \* 1.0;

}

integ[bb] = 1.0 \* h \* s \* arg \* 1.0;

bb = bb + 1;

}

break;

case 7: {

double s = 1.0 \* (log(a) + log(b)) / 2.0;

double h = 1.0 \* (b - a) / n \* 1.0;

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

{

s += 1.0 \* log(a + i \* h) \* 1.0;

}

integ[bb] = 1.0 \* h \* s \* arg \* 1.0;

bb = bb + 1;

}

break;

}

}

void trap(int b, int a, int n, int a4, int arg)

{

double y = 0.0;

switch (a4)

{

case 1: {

//double dy;

//y = 0.0;

double dy = 1.0\*(b - a) / n \* 1.0;

y += 1.0 \* sin(a) + sin(b) \* 1.0; // тут не так

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

{

y += 2.0 \* (sin(a + dy \* i) \* 1.0);

}

trapp[bbb] = 1.0 \* (b - a) / (2.0 \* n) \* y \* arg \* 1.0;

bbb = bbb + 1;

//integ[0] = 12;

// trapp[0] = 22;

}

break;

case 2: {

// double y, dy;

double dy =1.0\* (b - a) / n \* 1.0;

y += 1.0 \* cos(a) + cos(b) \* 1.0;

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

{

y += 2.0 \* cos(a + dy \* i) \* 1.0;

}

trapp[bbb] = 1.0 \* (b - a) / (2.0 \* n) \* y \* arg \* 1.0;

bbb = bbb + 1;

}

break;

case 3: {

//double y, dy;

double dy = 1.0\*(b - a) / n \* 1.0;

y += 1.0 \* sin(a)/cos(a) + sin(b)/cos(b) \* 1.0;

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

{

y += 2.0 \* sin(a + dy \* i)/cos(a+dy\*i) \* 1.0;

}

trapp[bbb] = 1.0 \* (b - a) / (2.0 \* n) \* y \* arg \* 1.0;

bbb = bbb + 1;

}

break;

case 4: {

//double y, dy;

double dy =1.0\* (b - a) / n \* 1.0;

y += 1.0 \* 1 / tan(a) + 1/tan(b);//1.0 \* cos(a)/sin(a) + cos(b)/sin(b) \* 1.0;

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

{

y += 2.0 \* 1 / tan(a + dy \* i);//2.0 \* cos(a+ dy \* i)/sin(a + dy \* i) \* 1.0;

}

trapp[bbb] = 1.0 \* (b - a) / (2.0 \* n) \* y \* arg \* 1.0;

bbb = bbb + 1;

}

break;

case 5: {

//double y, dy;

double dy =1.0\* (b - a) / n \* 1.0;

y += 1.0 \* a + b \* 1.0;

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

{

y += 2.0 \* (a + dy \* i) \* 1.0;

}

trapp[bbb] = 1.0 \* (b - a) / (2.0 \* n) \* y \* arg \* 1.0;

bbb = bbb + 1;

}

break;

case 6: {

// double y, dy;

double dy = 1.0\*(b - a) / n \* 1.0;

y += 1.0 \* a\*a + b\*b \* 1.0;

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

{

y += 2.0 \* (a + dy \* i)\*(a+dy\*i) \* 1.0;

}

trapp[bbb] = 1.0 \* (b - a) / (2.0 \* n) \* y \* arg \* 1.0;

bbb = bbb + 1;

}

break;

case 7: {

// double y, dy;

double dy =1.0\* (b - a) / n \* 1.0;

y += 1.0 \* log(a) + log(b) \* 1.0;

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

{

y += 2.0 \* log(a + dy \* i) \* 1.0;

}

trapp[bbb] = 1.0 \* (b - a) / (2.0 \* n) \* y \* arg \* 1.0;

bbb = bbb + 1;

}

break;

}

}

int CALLBACK wWinMain(HINSTANCE hInstance, HINSTANCE, PWSTR szCmdLine, int nCmdShow)

{

static const UINT MY\_DATA\_MSG{WM\_USER + 1000};

static MSG msg{};// Информация о сообщении

static HWND hwnd{}, hButton{}, hButton1{}, hChildWnd{}, hChildEdit1{}, hChildEdit2{}, hChildEdit3{}, hChildEdit4{};//Хранение дискриптора окна

WNDCLASSEX wc{sizeof(WNDCLASSEX)};

wc.cbClsExtra = 0;

wc.cbWndExtra = 0;

wc.hbrBackground = reinterpret\_cast<HBRUSH>(GetStockObject(WHITE\_BRUSH));

wc.hCursor = LoadCursor(nullptr, IDC\_ARROW);

wc.hIcon = LoadIcon(nullptr, IDI\_APPLICATION);

wc.hIconSm = LoadIcon(nullptr, IDI\_APPLICATION);

wc.hInstance = hInstance;

wc.lpfnWndProc = [](HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)->LRESULT //Процедура обработки сообщений

{

switch (uMsg)

{

case WM\_CREATE:

{

HWND hText = CreateWindow(TEXT("STATIC"), TEXT("Program for calculating the integral"),

WS\_VISIBLE | WS\_CHILD,

75, 10, 230, 50,

hWnd, nullptr, nullptr, nullptr

);

HWND hButton = CreateWindow(

L"BUTTON",

L"Result",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

50, 100, 300, 30, hWnd, reinterpret\_cast<HMENU>(39), nullptr, nullptr

);

HWND hButton1 = CreateWindow(

L"BUTTON",

L"Input ",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

50, 50, 300, 30, hWnd, reinterpret\_cast<HMENU>(12), nullptr, nullptr

);

}

return 0;

case WM\_COMMAND:

{

switch (LOWORD(wParam))

{

case 12:

{

//EnableWindow(hButton, False);

// hButton->Enabled = false;

//SetClassLong(hWnd, GCL\_STYLE, GetClassLong(hButton, GCL\_STYLE) | CS\_NOCLOSE);

if (okno == false && okno2==false) {

okno=true;

aa = ""; n = ""; a = ""; b = "";

s = 0; h = 0; I = 0;

ZeroMemory(znak,20);

ZeroMemory(buff, 200);

for (int i = 0; i > 20; i++)

integ[i] = 0;

for (int i = 0; i > 20; i++)

trapp[i] = 0;

for (int i = 0; i > 20; i++)

arg[i] = 0;

bb = 0; bbb = 0;

ii = 0; iii = 0; iiii = 0;

cc = "";

kek = 0;

k = ""; kk = "";

first = 0; second = 0;

ZeroMemory(res2, 10);

ZeroMemory(res1, 10);

a1 = 0; a2 = 0; a3 = 0;

wwe = 0;

fflag = false;

vv = ' ';

//okno = false;

del = 0;

kyk = false;

ZeroMemory(buf, 200);

//for (int i = 0; i < razmer + 1; i++)

//buf[i] = ' ';

razmer = 0;

aa = "";

bb = 0;

fflag = false;

//if (hChildWnd) DestroyWindow(hChildWnd);

const auto [flag, hChild] = AddWindow(L"MyAppChildClass", L"Input", hWnd, [](HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)->LRESULT {

switch (uMsg)

{

case WM\_CREATE:

{

HWND hButton = CreateWindow(

L"BUTTON",

L"Enter",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

115, 200, 200, 30, hWnd, reinterpret\_cast<HMENU>(122), nullptr, nullptr

);

HWND hEdit1 = CreateWindow(

L"Edit",

L"",

WS\_CHILD | WS\_BORDER | WS\_VISIBLE,

5, 0, 90, 20, hWnd, nullptr, nullptr, nullptr

);

hChildEdit1 = hEdit1;

HWND hEdit2 = CreateWindow(

L"Edit",

L"",

WS\_CHILD | WS\_BORDER | WS\_VISIBLE,

5, 25, 90, 20, hWnd, nullptr, nullptr, nullptr

);

hChildEdit2 = hEdit2;

HWND hEdit3 = CreateWindow(

L"Edit",

L"",

WS\_CHILD | WS\_BORDER | WS\_VISIBLE,

5, 50, 90, 20, hWnd, nullptr, nullptr, nullptr

);

hChildEdit3 = hEdit3;

HWND hEdit4 = CreateWindow(

L"Edit",

L"",

WS\_CHILD | WS\_BORDER | WS\_VISIBLE | ES\_READONLY,

5, 75, 300, 20, hWnd, reinterpret\_cast<HMENU>(14), nullptr, nullptr

);

hChildEdit4 = hEdit4;

HWND hText1 = CreateWindow(TEXT("STATIC"), TEXT("Upper limit"),

WS\_VISIBLE | WS\_CHILD,

100, 0, 120, 20,

hWnd, nullptr, nullptr, nullptr

);

HWND hText2 = CreateWindow(TEXT("STATIC"), TEXT("Lower limit"),

WS\_VISIBLE | WS\_CHILD,

100, 25, 120, 20,

hWnd, nullptr, nullptr, nullptr

);

HWND hText3 = CreateWindow(TEXT("STATIC"), TEXT("Number of splits"),

WS\_VISIBLE | WS\_CHILD,

100, 50, 120, 20,

hWnd, nullptr, nullptr, nullptr

);

HWND hText4 = CreateWindow(TEXT("STATIC"), TEXT("Definite integral"),

WS\_VISIBLE | WS\_CHILD,

310, 75, 120, 20,

hWnd, nullptr, nullptr, nullptr

);

HWND hButton1 = CreateWindow( //Кнопки для интегрирования

L"BUTTON",

L"sin(x)",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

5, 100, 50, 30, hWnd, reinterpret\_cast<HMENU>(1), nullptr, nullptr

);

HWND hButton2 = CreateWindow(

L"BUTTON",

L"cos(x)",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

60, 100, 50, 30, hWnd, reinterpret\_cast<HMENU>(2), nullptr, nullptr

);

HWND hButton3 = CreateWindow(

L"BUTTON",

L"tg(x)",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

115, 100, 50, 30, hWnd, reinterpret\_cast<HMENU>(3), nullptr, nullptr

);

HWND hButton4 = CreateWindow(

L"BUTTON",

L"ctg(x)",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

170, 100, 50, 30, hWnd, reinterpret\_cast<HMENU>(4), nullptr, nullptr

);

HWND hButton5 = CreateWindow(

L"BUTTON",

L"x",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

225, 100, 50, 30, hWnd, reinterpret\_cast<HMENU>(5), nullptr, nullptr

);

HWND hButton6 = CreateWindow(

L"BUTTON",

L"(x^2)",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

280, 100, 50, 30, hWnd, reinterpret\_cast<HMENU>(6), nullptr, nullptr

);

HWND hButton7 = CreateWindow(

L"BUTTON",

L"ln(x)",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

335, 100, 50, 30, hWnd, reinterpret\_cast<HMENU>(7), nullptr, nullptr

);

HWND hButton9 = CreateWindow(

L"BUTTON",

L"+",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

5, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(9), nullptr, nullptr

);

HWND hButton10 = CreateWindow(

L"BUTTON",

L"-",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

40, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(10), nullptr, nullptr

);

HWND hButton11 = CreateWindow(

L"BUTTON",

L"1",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

75, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(11), nullptr, nullptr

);

HWND hButton12 = CreateWindow(

L"BUTTON",

L"2",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

110, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(12), nullptr, nullptr

);

HWND hButton13 = CreateWindow(

L"BUTTON",

L"3",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

145, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(13), nullptr, nullptr

);

HWND hButton14 = CreateWindow(

L"BUTTON",

L"4",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

180, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(14), nullptr, nullptr

);

HWND hButton15 = CreateWindow(

L"BUTTON",

L"5",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

215, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(15), nullptr, nullptr

);

HWND hButton16 = CreateWindow(

L"BUTTON",

L"6",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

250, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(16), nullptr, nullptr

);

HWND hButton17 = CreateWindow(

L"BUTTON",

L"7",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

285, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(17), nullptr, nullptr

);

HWND hButton18 = CreateWindow(

L"BUTTON",

L"8",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

320, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(18), nullptr, nullptr

);

HWND hButton19 = CreateWindow(

L"BUTTON",

L"9",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

355, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(19), nullptr, nullptr

);

HWND hButton21 = CreateWindow(

L"BUTTON",

L"0",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

390, 135, 30, 30, hWnd, reinterpret\_cast<HMENU>(21), nullptr, nullptr

);

/\* HWND hButton20 = CreateWindow(

L"BUTTON",

L"←",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

390, 100, 40, 65, hWnd, reinterpret\_cast<HMENU>(20), nullptr, nullptr

);\*/

}

return 0;

case WM\_COMMAND:

{

switch (LOWORD(wParam))

{

case 122:

{

aa = "";

iii = 0;

bb = 0;

cc = ' ';

ii = 0;

if (buf[0] != '-')

{

for (int i = razmer; i >= 0; i--)

{

buf[i + 1] = buf[i];

}

buf[0] = '+';

}

for (int i = 0; i < razmer; i++) //Обработчик знаков функции

{

if (buf[i] == '-' || buf[i] == '+')

{

znak[ii] = buf[i];

ii = ii + 1;

}

}

for (int i = 0; i < razmer; i++) //Обработчик аргументов функций

{

if (buf[i] == '-' || buf[i] == '+')

{

if (buf[i + 1] == '1' || buf[i + 1] == '2' || buf[i + 1] == '3' || buf[i + 1] == '4' || buf[i + 1] == '5' || buf[i + 1] == '6' || buf[i + 1] == '7' || buf[i + 1] == '8' || buf[i + 1] == '9' || buf[i+1]=='0')

{

cc = buf[i + 1];

if (buf[i + 2] == '1' || buf[i + 2] == '2' || buf[i + 2] == '3' || buf[i + 2] == '4' || buf[i + 2] == '5' || buf[i + 2] == '6' || buf[i + 2] == '7' || buf[i + 2] == '8' || buf[i + 2] == '9' || buf[i + 1] == '0')

{

cc += buf[i + 2];

if (buf[i + 3] == '1' || buf[i + 3] == '2' || buf[i + 3] == '3' || buf[i + 3] == '4' || buf[i + 3] == '5' || buf[i + 3] == '6' || buf[i + 3] == '7' || buf[i + 3] == '8' || buf[i + 3] == '9' || buf[i + 1] == '0')

{

cc += buf[i + 3];

if (buf[i + 4] == '1' || buf[i + 4] == '2' || buf[i + 4] == '3' || buf[i + 4] == '4' || buf[i + 4] == '5' || buf[i + 4] == '6' || buf[i + 4] == '7' || buf[i + 4] == '8' || buf[i + 4] == '9' || buf[i + 1] == '0')

{

cc += buf[i + 4];

arg[iii] = stoi(cc);

iii = iii + 1;

}

else

{

arg[iii] = stoi(cc);

iii = iii + 1;

}

}

else

{

arg[iii] = stoi(cc);

iii = iii + 1;

}

}

else

{

arg[iii] = stoi(cc);

iii = iii + 1;

}

}

else

{

arg[iii] = 1;

iii = iii + 1;

}

}

}

try {

TCHAR tcA[4];

TCHAR tcB[4];

TCHAR tcN[5];

//char kk[2];

string ss, sss, ssss;

GetWindowText(hChildEdit1, tcA, 4);

GetWindowText(hChildEdit2, tcB, 4);

GetWindowText(hChildEdit3, tcN, 5);

for (int i = 0; i < 3; i++)

sss += static\_cast<TCHAR>(tcA[i]);

a1 = stoi(sss);

for (int i = 0; i < 3; i++)

ssss += static\_cast<TCHAR>(tcB[i]);

a2 = stoi(ssss);

for (int i = 0; i < 4; i++)

ss += static\_cast<TCHAR>(tcN[i]);

a3 = stoi(ss);

fflag = true;

if (a1 < 0 || a1>99 || a2 < 0 || a2>99)

{

MessageBox(hWnd, L"The limits of integration shoud be within [0..99]", L"Out of range", MB\_OK | MB\_ICONSTOP);

} else

if (a3 < 1 || a3>999)

{

MessageBox(hWnd, L"The number of splits shoud be within [1..999]", L"Out of range", MB\_OK | MB\_ICONSTOP);

}

else

if (a2 > a1)

{

MessageBox(hWnd, L"The upper limit of integration must be greater than or equal to the lower limit", L"Wrong integration limits", MB\_OK | MB\_ICONSTOP);

}

// else

/\*if (buf[3] == ' ')

{

MessageBox(hWnd, L"All fields should be fill in", L"ppp", MB\_OK | MB\_ICONSTOP);

}\*/ else kyk = true;

fflag = true;

}

catch (invalid\_argument) {

MessageBox(hWnd, L"All fields must be filled in correctly", L"Incorrect input", MB\_OK | MB\_ICONSTOP);

}

catch (out\_of\_range) {

MessageBox(hWnd, L"The limits of integration shoud be within [1..99]", L"Out of range", MB\_OK | MB\_ICONSTOP);

}

if (kyk == true) {

if (a1 != 0 || a2 != 0) {

for (int i = 0; i < razmer + 1; i++)

{

if (buf[i] == 's' && buf[i + 1] == 'i')

{

pryamoygolnik(a1, a2, a3, 1, arg[kek]);

trap(a1, a2, a3, 1, arg[kek]);

kek = kek + 1;

}

else

if (buf[i] == 'c' && buf[i + 1] == 'o')

{

pryamoygolnik(a1, a2, a3, 2, arg[kek]);

trap(a1, a2, a3, 2, arg[kek]);

kek = kek + 1;

}

else if (buf[i] == 't' && buf[i + 1] == 'g' && buf[i - 1] != 'c')

{

pryamoygolnik(a1, a2, a3, 3, arg[kek]);

trap(a1, a2, a3, 3, arg[kek]);

kek = kek + 1;

}

else

if (buf[i] == 'c' && buf[i + 1] == 't')

{

pryamoygolnik(a1, a2, a3, 4, arg[kek]);

trap(a1, a2, a3, 4, arg[kek]);

kek = kek + 1;

}

else

if (buf[i] == 'x' && buf[i + 1] != ')' && buf[i + 1] != '^')

{

pryamoygolnik(a1, a2, a3, 5, arg[kek]);

trap(a1, a2, a3, 5, arg[kek]);

kek = kek + 1;

}

else

if (buf[i] == 'x' && buf[i + 1] == '^')

{

pryamoygolnik(a1, a2, a3, 6, arg[kek]);

trap(a1, a2, a3, 6, arg[kek]);

kek = kek + 1;

}

else

if (buf[i] == 'l' && buf[i + 1] == 'n')

{

pryamoygolnik(a1, a2, a3, 7, arg[kek]);

trap(a1, a2, a3, 7, arg[kek]);

kek = kek + 1;

}

}

first = integ[0];

second = trapp[0];

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

{

if (znak[i] == '-') //Предусмотреть для первого отрицательного аргумента

{

first = first - integ[i];

second = second - trapp[i];

}

else

{

first = first + integ[i];

second = second + trapp[i];

}

}

// fflag = true;

if (fflag == true)

{

DestroyWindow(hChildWnd);

okno = false;

}

}

}

}

break;

case 1:

{

//if (razmer==0)

if (razmer == 0 || aa[razmer - 1] == '-' || aa[razmer - 1] == '+' || isdigit(aa[razmer - 1]))

{

razmer = razmer + 6;

aa += "sin(x)";

del = 6;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer + 1);

}

else Beep(600, 300);

}

break;

case 2:

{

if (razmer == 0 || aa[razmer - 1] == '-' || aa[razmer - 1] == '+' || isdigit(aa[razmer - 1]))

{

razmer = razmer + 6;

aa += "cos(x)";

del = 6;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer + 1);

}

else Beep(200, 300);

}

break;

case 3:

{

if (razmer == 0 || aa[razmer - 1] == '-' || aa[razmer - 1] == '+' || isdigit(aa[razmer - 1]))

{

razmer = razmer + 5;

aa += "tg(x)";

del = 5;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer + 1);

}

else Beep(400, 800);

}

break;

case 4:

{

if (razmer == 0 || aa[razmer - 1] == '-' || aa[razmer - 1] == '+' || isdigit(aa[razmer - 1]))

{

razmer = razmer + 6;

del = 6;

aa += "ctg(x)";

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer + 1);

}

else Beep(700, 400); }

break;

case 5:

{

if (razmer == 0 || aa[razmer - 1] == '-' || aa[razmer - 1] == '+' || isdigit(aa[razmer - 1]))

{

razmer = razmer + 1;

aa += "x";

del = 1;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer + 1);

}

else Beep(400, 890);

}

break;

case 6:

{

if (razmer == 0 || aa[razmer - 1] == '-' || aa[razmer - 1] == '+' || isdigit(aa[razmer - 1]))

{

razmer = razmer + 5;

aa += "(x^2)";

del = 5;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer + 1);

}

else Beep(500, 300);

}

break;

case 7:

{

if (razmer == 0 || aa[razmer - 1] == '-' || aa[razmer - 1] == '+' || isdigit(aa[razmer - 1]))

{

razmer = razmer + 5;

aa += "ln(x)";

del = 5;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer + 1);

}

else Beep(600, 200);

}

break;

case 9: //Начало цифр и знаков

{

//cifra("+", hChildEdit4);

if (razmer != 0)

if (aa[razmer - 1] == ')' || aa[razmer - 1] == 'x') {

razmer = razmer + 1;

aa += '+';

del = 1;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer);

}

else Beep(500, 700);

}

break;

case 10:

{

//cifra("-", hChildEdit4);

if (razmer != 0)

if (aa[razmer - 1] == ')' || aa[razmer - 1] == 'x') {

razmer = razmer + 1;

aa += '-';

del = 1;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer);

}

else Beep(500, 700);

}

break;

case 11:

{

cifra("1", hChildEdit4);

}

break;

case 12:

{

cifra("2", hChildEdit4);

}

break;

case 13:

{

cifra("3", hChildEdit4);

}

break;

/\* case 14:

{

cifra("4", hChildEdit4);

}

break;\*/

case 15:

{

cifra("5", hChildEdit4);

}

break;

case 16:

{

cifra("6", hChildEdit4);

}

break;

case 17:

{

cifra("7", hChildEdit4);

}

break;

case 18:

{

cifra("8", hChildEdit4);

}

break;

case 19:

{

cifra("9", hChildEdit4);

}

break;

case 21:

{

cifra("0", hChildEdit4);

}

break;

/\* case 20:

{

if (razmer > 0)

{

switch (del)

{

case 1:

{

buf[razmer] = '\0';

aa[razmer] = '\0';

razmer = razmer - 1;

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

{

buf[i] = aa[i];

}

SetWindowText(hChildEdit4, buf);

GetWindowText(hChildEdit4, buf, razmer);

}

break;

case 3:

{

}

break;

case 5:

{

}

break;

case 6:

{

}

break;

}

}

else Beep(329, 500);

}

break;\*/

}

// DestroyWindow(hChildWnd);

}

case MY\_DATA\_MSG:

{

}

return 0;

}

return DefWindowProc(hWnd, uMsg, wParam, lParam);

});

hChildWnd = hChild;

// DestroyWindow(hChildWnd);

}

}

break;

case 39:

{

if (razmer == 0)

MessageBox(hWnd, L"Data input is empty", L"Input is empty", MB\_OK | MB\_ICONINFORMATION); else

if (okno2==false){

const auto [flag, hChild] = AddWindow(L"MyAppChildClass", L"Result", hWnd, [](HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)->LRESULT {

switch (uMsg)

{

case WM\_CREATE:

{

HWND hText1 = CreateWindow(TEXT("STATIC"), TEXT("Rectangle method"),

WS\_VISIBLE | WS\_CHILD,

140, 0, 150, 20,

hWnd, nullptr, nullptr, nullptr

);

HWND hText2 = CreateWindow(TEXT("STATIC"), TEXT("Trapezoid method"),

WS\_VISIBLE | WS\_CHILD,

140, 100, 150, 20,

hWnd, nullptr, nullptr, nullptr

);

HWND hButton = CreateWindow(

L"BUTTON",

L"Close",

WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON,

100, 200, 200, 30, hWnd, reinterpret\_cast<HMENU>(399), nullptr, nullptr

);

HWND hEdit1 = CreateWindow(

L"Edit",

L"",

WS\_CHILD | WS\_BORDER | WS\_VISIBLE | ES\_READONLY,

150, 50, 100, 20, hWnd, nullptr, nullptr, nullptr

);

hChildEdit1 = hEdit1;

HWND hEdit2 = CreateWindow(

L"Edit",

L"",

WS\_CHILD | WS\_BORDER | WS\_VISIBLE | ES\_READONLY,

150, 150, 100, 20, hWnd, nullptr, nullptr, nullptr

);

hChildEdit2 = hEdit2;

}

vv = static\_cast<char>(znak[0]);

okno2 = true;

//SetWindowText(hChildEdit1, to\_wstring(first).c\_str());

SetWindowText(hChildEdit1, to\_wstring(first).c\_str());

SetWindowText(hChildEdit2, to\_wstring(second).c\_str());

return 0;

case WM\_COMMAND:

{

switch (LOWORD(wParam))

{

case 399:

{

okno2 = false;

DestroyWindow(hChildWnd);

}

}

}

case MY\_DATA\_MSG:

{

}

return 0;

}

return DefWindowProc(hWnd, uMsg, wParam, lParam);

});

hChildWnd = hChild;

}

}

break;

}

}

return 0;

case WM\_DESTROY:

{

PostQuitMessage(EXIT\_SUCCESS);

}

return 0;

}

return DefWindowProc(hWnd, uMsg, wParam, lParam);

};

wc.lpszClassName = L"MyAppClass";

wc.lpszMenuName = nullptr;

wc.style = CS\_VREDRAW | CS\_HREDRAW;

if (!RegisterClassEx(&wc)) //Регистрация окна в системе

return EXIT\_FAILURE;

if (hwnd = CreateWindow(wc.lpszClassName, L"Calculation of the integral", WS\_OVERLAPPEDWINDOW & ~WS\_THICKFRAME & ~WS\_MAXIMIZEBOX,

(GetSystemMetrics(SM\_CXSCREEN) - 400) / 2,

(GetSystemMetrics(SM\_CYSCREEN) - 400) / 2,

400, 200, nullptr, nullptr, wc.hInstance, nullptr); hwnd == INVALID\_HANDLE\_VALUE)

return EXIT\_FAILURE;

ShowWindow(hwnd, nCmdShow);

UpdateWindow(hwnd);

while (GetMessage(&msg, nullptr, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return static\_cast<int>(msg.wParam);

}

**Экранные формы**

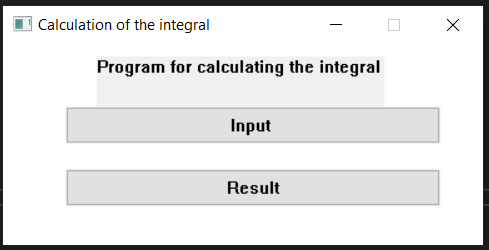


Рисунок 1 - Главное окно программы

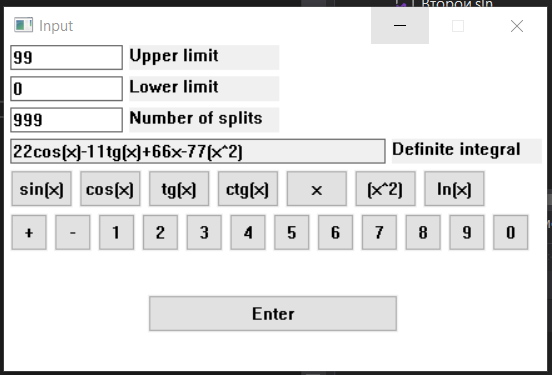


Рисунок 2 – Дочернее окно для ввода данных

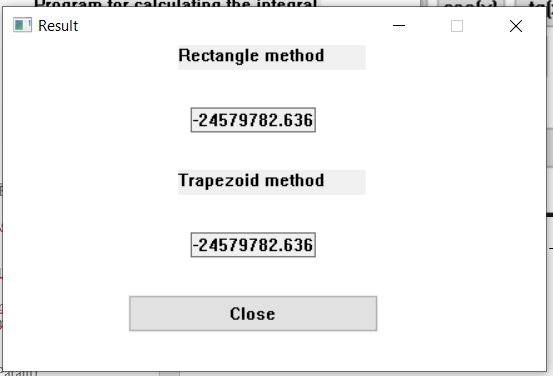


Рисунок 3 – Дочернее окно для вывода результата

**Вывод**

В ходе выполнения лабораторной работы была написана программа с использованием функций WinAPI. В программе была реализована возможность расчёта интеграла двумя методами: методом прямоугольников и методом трапеций.

Особенность лабораторной работы заключалась в том, что помимо главного окна нужно было реализовать два дочерних окна. В одном из них осуществляется ввод информации, а во втором вывод результата расчётов.

Перед началом выполнения данной лабораторной работы было решено, что функция интеграла будет собираемой, т.е. пользователь может сам, при помощи «кнопок», задавать нужный интеграл, конечно же, с некоторыми ограничениями. Были добавлены такие стандартные функции, как sin(x), cos(x), tg(x), ctg(x), ln(x), x и (x^2). Помимо этого, пользователю разрешено вводить целочисленные аргументы для каждой функции. Неотъемлемой частью расчёта интеграла при помощи какого-либо метода являются пределы интегрирования и количество разбиений. Все эти данные тоже вводятся вручную пользователем. Далее, при нажатии кнопки «Enter», происходит расчёт введённого интеграла и его вывод во второе дочернее окно.