ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра «Информационные системы»

О Т Ч Е Т

о выполнении лабораторной работы № 1

по дисциплине «Проектирование человеко-машинного интерфейса»

по теме: «Реализация простого Windows - приложения»

Выполнили: Козлов А.А., Ноздря О.Д. Шифр: 140211, 140213

ИПАИТ

Направление / специальность: 09.03.04 «Программная инженерия»

Группа: 41-ПГ

Преподаватель Олькина Е.В.

Отчет защищен с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_» Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

Орел, 2017

**Программный код**

// WindowsProject1.cpp: определяет точку входа для приложения.

//

#include "stdafx.h"

#include "WindowsProject1.h"

#include <string>

#define MAX\_LOADSTRING 100

// Глобальные переменные:

HINSTANCE hInst; // текущий экземпляр

WCHAR szTitle[MAX\_LOADSTRING]; // Текст строки заголовка

WCHAR szWindowClass[MAX\_LOADSTRING]; // имя класса главного окна

TCHAR greeting[] = \_T("Test App");

// Отправить объявления функций, включенных в этот модуль кода:

ATOM MyRegisterClass(HINSTANCE hInstance);

BOOL InitInstance(HINSTANCE, int);

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

HWND window = nullptr;

HWND child\_window = nullptr;

int APIENTRY wWinMain(\_In\_ HINSTANCE hInstance,

\_In\_opt\_ HINSTANCE hPrevInstance,

\_In\_ LPWSTR lpCmdLine,

\_In\_ int nCmdShow)

{

UNREFERENCED\_PARAMETER(hPrevInstance);

UNREFERENCED\_PARAMETER(lpCmdLine);

// TODO: разместите код здесь.

// Инициализация глобальных строк

LoadStringW(hInstance, IDS\_APP\_TITLE, szTitle, MAX\_LOADSTRING);

LoadStringW(hInstance, IDC\_WINDOWSPROJECT1, szWindowClass, MAX\_LOADSTRING);

MyRegisterClass(hInstance);

// Выполнить инициализацию приложения:

if (!InitInstance (hInstance, nCmdShow))

{

return FALSE;

}

HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_WINDOWSPROJECT1));

MSG msg;

// Цикл основного сообщения:

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

{

if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

}

return (int) msg.wParam;

}

//

// ФУНКЦИЯ: MyRegisterClass()

//

// НАЗНАЧЕНИЕ: регистрирует класс окна.

//

ATOM MyRegisterClass(HINSTANCE hInstance)

{

WNDCLASSEXW wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

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

wcex.lpfnWndProc = WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

wcex.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI\_WINDOWSPROJECT1));

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

wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW+1);

wcex.lpszMenuName = NULL; //MAKEINTRESOURCEW(IDC\_WINDOWSPROJECT1);

wcex.lpszClassName = szWindowClass;

wcex.hIconSm = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI\_SMALL));

return RegisterClassExW(&wcex);

}

//

// ФУНКЦИЯ: InitInstance(HINSTANCE, int)

//

// НАЗНАЧЕНИЕ: сохраняет обработку экземпляра и создает главное окно.

//

// КОММЕНТАРИИ:

//

// В данной функции дескриптор экземпляра сохраняется в глобальной переменной, а также

// создается и выводится на экран главное окно программы.

//

BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)

{

hInst = hInstance; // Сохранить дескриптор экземпляра в глобальной переменной

window = CreateWindowW(szWindowClass, greeting, WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, 0, CW\_USEDEFAULT, 0, nullptr, nullptr, hInstance, nullptr);

if (!window)

{

return FALSE;

}

ShowWindow(window, nCmdShow);

UpdateWindow(window);

return TRUE;

}

//

// ФУНКЦИЯ: WndProc(HWND, UINT, WPARAM, LPARAM)

//

// НАЗНАЧЕНИЕ: обрабатывает сообщения в главном окне.

//

// WM\_COMMAND — обработать меню приложения

// WM\_PAINT — отрисовать главное окно

// WM\_DESTROY — отправить сообщение о выходе и вернуться

//

//

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

int x, y;

POINT pt;

x = LOWORD(lParam); //узнаём координаты

y = HIWORD(lParam);

GetCursorPos(&pt);

switch (message)

{

case WM\_RBUTTONDOWN:

{

char buffer[64], buffer1[64];

sprintf\_s(buffer, "Window - X: %d, Y: %d\n", x, y);

sprintf\_s(buffer1, "Screen - X: %d, Y: %d", pt.x, pt.y);

strcat\_s(buffer, buffer1);

MessageBoxA(hWnd, buffer, "window coord", MB\_OK | MB\_ICONINFORMATION);

//MessageBoxA(hWnd, buffer1, "screen coord", MB\_OK | MB\_ICONINFORMATION);

}

break;

case WM\_LBUTTONDOWN:

{

if (child\_window == nullptr) {

child\_window = CreateWindowW(szWindowClass, greeting, WS\_OVERLAPPEDWINDOW | WS\_VISIBLE,

CW\_USEDEFAULT, 0, 300, 300, window, nullptr, nullptr, nullptr);

ShowWindow(child\_window, true);

UpdateWindow(child\_window);

}

else {

MoveWindow(child\_window, x, y, 300, 300, true);

}

}

break;

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

// TODO: Добавьте сюда любой код прорисовки, использующий HDC...

EndPaint(hWnd, &ps);

}

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

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

}

return 0;

}