МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационных систем

**ОТЧЕТ**

по лабораторной работе № 3

на тему: «Многооконное MDI приложение»

по дисциплине: «Проектирование пользовательского интерфейса»

Выполнили: Кожухова О.А., Шорин В.Д.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверили: Олькина Е.В., Лысков О.Э.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Орел, 2019 г.

**Задание на лабораторную работу:**

Реализовать windows приложение, которое в главном окне производит обработку нескольких текстовых документов используя интерфейс MDI. При изменении текста в любом из окон происходит изменения текста в других окнах.

**Выполнение работы:**

#include "windows.h"

#include <stdio.h>

char szMainClass[] = "Main Class";

char text[1024];

BOOL RegClass(WNDPROC, LPCTSTR, UINT);

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

HINSTANCE hInst;

HWND edit1;

HWND edit2;

int WINAPI WinMain(HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPSTR lpCmdLine,

int nCmdShow)

{

HWND hwnd;

MSG msg;

hInst = hInstance;

if (!RegClass(WndProc, szMainClass, COLOR\_WINDOW)) { return FALSE; }

hwnd = CreateWindow(szMainClass,

"Main Window",

WS\_OVERLAPPEDWINDOW | WS\_VISIBLE,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

0,

0,

hInst,

NULL);

ShowWindow(hwnd, SW\_SHOWDEFAULT);

if (!hwnd)

{

MessageBox(NULL, "Window Creation Failed.", "Error", MB\_OK | MB\_ICONERROR);

return FALSE;

}

HACCEL hAccel = LoadAccelerators(hInst, MAKEINTRESOURCE(0));

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

{

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

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

}

return msg.wParam;

}

BOOL RegClass(WNDPROC Proc, LPCTSTR szName, UINT brBackground)

{

WNDCLASS wc;

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

wc.cbClsExtra = 0;

wc.cbWndExtra = 0;

wc.lpfnWndProc = Proc;

wc.hInstance = hInst;

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

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

wc.hbrBackground = (HBRUSH)(brBackground + 1);

wc.lpszMenuName = (LPCTSTR)NULL;

wc.lpszClassName = szName;

return (RegisterClass(&wc) != 0);

}

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

{

switch (msg)

{

case WM\_CREATE:

{

RECT mainRect;

GetWindowRect(hWnd, &mainRect);

int width = mainRect.right - mainRect.left;

int height = mainRect.bottom - mainRect.top;

edit1 = CreateWindow("Edit",

"",

WS\_CHILD | WS\_VISIBLE | WS\_BORDER | ES\_MULTILINE | ES\_AUTOVSCROLL,

0,

0,

width / 2,

height,

hWnd,

0,

hInst,

NULL);

if (!edit1)

{

MessageBox(NULL, "Edit1 Creation Failed.", "Error", MB\_OK | MB\_ICONERROR);

return FALSE;

}

edit2 = CreateWindow("Edit",

"",

WS\_CHILD | WS\_VISIBLE | WS\_BORDER | ES\_MULTILINE | ES\_AUTOVSCROLL,

width / 2,

0,

width,

height,

hWnd,

NULL,

hInst,

NULL);

if (!edit2)

{ MessageBox(NULL, "Edit2 Creation Failed.", "Error", MB\_OK | MB\_ICONERROR);

return FALSE;

}

break;

}

case WM\_COMMAND:

{ if (GetFocus() == edit1) {

GetWindowText(edit1, text, 1024);

SetWindowText(edit2, text);

}

else {

GetWindowText(edit2, text, 1024);

SetWindowText(edit1, text);

}

break;

}

case WM\_DESTROY:

{ PostQuitMessage(0);

break;

}

default:

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

}

}