Міністерство освіти і науки України Харківський національний університет радіоелектроніки

Кафедра Інформатики

Звіт

з Практичної роботи №3

з дисципліни

«Операційні системи»

Виконав: Перевірив:

ITIHФ-20-1 Професор

Самченко С.О. Сінельнікова Т.Ф.

Харків – 2022

Стиль класу: заборонити команду «закрити» в системному меню, перерисовка вікна при зміні вертикального і горизонтального розміру;

* Тип завантажується іконки: IDI\_APPLICATION,
* Тип завантажується курсору: IDC\_HELP.
* Колір фону вікна: СВІТЛО СІРИЙ,
* Ім'я класу вікна: ім'я студента (англійською мовою).
* Стиль вікна:
  + а) вікно має заголовок і обрамляє рамку, у вікна є кнопка мінімізації, у вікна є системне меню;
  + б) WS\_OVERLAPPEDWINDOW. Позиція вікна: по лівому краю екрану шириною 500 і висотою 200 точок

За допомогою меню змінювати розміри вікна програми (3 стандартних розміру).

**framework.h:**

#pragma once

#include "targetver.h"

#define WIN32\_LEAN\_AND\_

#include <windows.h>

#include <stdlib.h>

#include <malloc.h>

#include <memory.h>

#include <tchar.h>

**Hook.h:**

#pragma once

#include "resource.h"

**Resource.h:**

//{{NO\_DEPENDENCIES}}

// Включаемый файл, созданный в Microsoft Visual C++.

// Используется Hook.rc

#define IDS\_APP\_TITLE 103

#define IDR\_MAINFRAME 128

#define IDD\_HOOK\_DIALOG 102

#define IDD\_ABOUTBOX 103

#define IDM\_ABOUT 104

#define IDM\_EXIT 105

#define IDI\_HOOK 107

#define IDI\_SMALL 108

#define IDC\_HOOK 109

#define IDC\_MYICON 2

#ifndef IDC\_STATIC

#define IDC\_STATIC -1

#endif

// Следующие стандартные значения для новых объектов

//

#ifdef APSTUDIO\_INVOKED

#ifndef APSTUDIO\_READONLY\_SYMBOLS

#define \_APS\_NO\_MFC 130

#define \_APS\_NEXT\_RESOURCE\_VALUE 129

#define \_APS\_NEXT\_COMMAND\_VALUE 32771

#define \_APS\_NEXT\_CONTROL\_VALUE 1000

#define \_APS\_NEXT\_SYMED\_VALUE 110

#endif

#endif

**targetver.h:**

#pragma once

#include <SDKDDKVer.h>

**Source.cpp:**

#include "framework.h"

#include "PZ3.h"

#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

HMENU pMenu;

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

ATOM MyRegisterClass(HINSTANCE hInstance);

BOOL InitInstance(HINSTANCE, int);

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

INT\_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);

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\_LABA3VARIANT17, szWindowClass, MAX\_LOADSTRING);

MyRegisterClass(hInstance);

// Выполнить инициализацию приложения:

if (!InitInstance (hInstance, nCmdShow))

{

return FALSE;

}

HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_LABA3VARIANT17));

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()

//

// ЦЕЛЬ: Регистрирует класс окна.

//

ATOM MyRegisterClass(HINSTANCE hInstance)

{

WNDCLASSEXW wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

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(nullptr, IDI\_APPLICATION);

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

wcex.hbrBackground = reinterpret\_cast<HBRUSH>(GetStockObject(LTGRAY\_BRUSH));

wcex.lpszMenuName = MAKEINTRESOURCEW(IDC\_LABA3VARIANT17);

wcex.lpszClassName = szWindowClass;

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

return RegisterClassExW(&wcex);

}

//

// ФУНКЦИЯ: InitInstance(HINSTANCE, int)

//

// ЦЕЛЬ: Сохраняет маркер экземпляра и создает главное окно

//

// КОММЕНТАРИИ:

//

// В этой функции маркер экземпляра сохраняется в глобальной переменной, а также

// создается и выводится главное окно программы.

//

BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)

{

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

HWND hWnd = CreateWindowW(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,

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

if (!hWnd)

{

return FALSE;

}

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

return TRUE;

}

//

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

//

// ЦЕЛЬ: Обрабатывает сообщения в главном окне.

//

// WM\_COMMAND - обработать меню приложения

// WM\_PAINT - Отрисовка главного окна

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

//

//

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

{

RECT rect = { NULL };

switch (message)

{

case WM\_CREATE:

pMenu = GetSystemMenu(hWnd, FALSE); // Получаем системное меню.

EnableMenuItem(pMenu, SC\_CLOSE, MF\_BYCOMMAND | MF\_GRAYED);

break;

case WM\_COMMAND:

{

int wmId = LOWORD(wParam);

// Разобрать выбор в меню:

switch (wmId)

{

case IDM\_SIZE\_ONE :

GetWindowRect(hWnd, &rect);

SetWindowPos(hWnd, HWND\_TOP, rect.left, rect.top, 720, 576, SWP\_SHOWWINDOW);

break;

case IDM\_SIZE\_TWO:

GetWindowRect(hWnd, &rect);

SetWindowPos(hWnd, HWND\_TOP, rect.left, rect.top, 800, 600, SWP\_SHOWWINDOW);

break;

case IDM\_SIZE\_THREE:

GetWindowRect(hWnd, &rect);

SetWindowPos(hWnd, HWND\_TOP, rect.left, rect.top, 1280, 1024, SWP\_SHOWWINDOW);

break;

case IDM\_ABOUT:

DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);

break;

case IDM\_EXIT:

DestroyWindow(hWnd);

break;

default:

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

}

}

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;

}

// Обработчик сообщений для окна "О программе".

INT\_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)

{

UNREFERENCED\_PARAMETER(lParam);

switch (message)

{

case WM\_INITDIALOG:

return (INT\_PTR)TRUE;

case WM\_COMMAND:

if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)

{

EndDialog(hDlg, LOWORD(wParam));

return (INT\_PTR)TRUE;

}

break;

}

return (INT\_PTR)FALSE;

}

