МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт КНТ

Факультет ФИСП

Кафедра ПИ им. Л.П. Фельдмана

Лабораторная работа №1

Тема: «Работа с окнами»

Курс: Системное программирование

Выполнил

ст. гр. ПИ-18б

Моргунов А.Г.

Проверил

Филипишин Д.А.

Донецк – 2021

**Задание**

5. Программа должна содержать функцию WinMain() и оконную процедуру. В функции WinMain() выполнить регистрацию оконного класса, создание окна и запуск цикла обработки сообщений. В оконной процедуре обработать сообщения от окна. Также в функции WinMain() создать подчинённое окно на основе того же класса, что и главное, и расположить его таким образом, что бы левый верхний угол подчинённого окна совпал с левым верхним углом главного. При нажатии на правую кнопку мыши перемесить подчинённое окно в соседний по ходу часовой стрелки угол главного окна.

**Код**

#define UNICODE

#include <windows.h>

#include <tchar.h>

#include <unordered\_map>

#include <iostream>

#include <functional>

struct WindowSize {

int width;

int height;

};

WindowSize GetWindowSize(HWND window\_handle) {

RECT window\_rect;

GetWindowRect(window\_handle, &window\_rect);

WindowSize result;

result.width = window\_rect.right - window\_rect.left;

result.height = window\_rect.bottom - window\_rect.top;

return result;

}

WindowSize GetClientSize(HWND window\_handle) {

RECT window\_rect;

GetClientRect(window\_handle, &window\_rect);

WindowSize result;

result.width = window\_rect.right - window\_rect.left;

result.height = window\_rect.bottom - window\_rect.top;

return result;

}

using ProcHandler = std::function<void(HWND hWnd, UINT msg,

WPARAM wParam, LPARAM lParam)>;

HWND hWnd;

HWND child;

std::unordered\_map<HWND, ProcHandler> handlers;

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

TCHAR WinName[] = \_T(L"MainFrame");

int WINAPI WinMain(HINSTANCE This, // Дескриптор текущего приложения

HINSTANCE Prev, // В современных системах всегда 0

LPSTR cmd, // Командная строка

int mode) // Режим отображения окна

{

// Дескриптор главного окна программы

MSG msg; // Структура для хранения сообщения

WNDCLASS wc; // Класс окна

memset(&wc, 0, sizeof(wc));

// Определение класса окна

wc.hInstance = This;

wc.lpszClassName = WinName; // Имя класса окна

wc.lpfnWndProc = WndProc; // Функция окна

wc.style = CS\_HREDRAW | CS\_VREDRAW; // Стиль окна

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

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

wc.lpszMenuName = NULL; // Нет меню

wc.cbClsExtra = 0; // Нет дополнительных данных класса

wc.cbWndExtra = 0; // Нет дополнительных данных окна

// Заполнение окна белым цветом

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

if(!RegisterClass(&wc)) return -100; // Регистрация класса окна

// Создание окна

hWnd = CreateWindow(WinName, // Имя класса окна

\_T(L"Каркас Windows-приложения"), // Заголовок окна

WS\_OVERLAPPEDWINDOW, // Стиль окна

100,// x

100,// y Размеры окна

500,// Width

500,// Height

HWND\_DESKTOP, // Дескриптор родительского окна

NULL, // Нет меню

This, // Дескриптор приложения

NULL); // Дополнительной информации нет

child = CreateWindow(WinName,

\_T(L"Child Window"),

WS\_CAPTION | WS\_CHILD,

0,

0,

150,

100,

hWnd,

NULL,

This,

NULL);

ShowWindow(hWnd, mode); //Показать окно

ShowWindow(child, mode);

handlers[hWnd] = [](HWND hWnd, UINT msg,

WPARAM wParam, LPARAM lParam) {

};

handlers[child] = [Parent = hWnd](HWND hWnd, UINT msg,

WPARAM wParam, LPARAM lParam){

WindowSize parentSize = GetClientSize(Parent);

WindowSize mySize = GetWindowSize(hWnd);

static int status = 0;

switch (status) {

case 0:

MoveWindow(hWnd, parentSize.width - mySize.width, 0,

mySize.width, mySize.height, TRUE);

break;

case 1:

MoveWindow(hWnd, parentSize.width - mySize.width, parentSize.height - mySize.height,

mySize.width, mySize.height, TRUE);

break;

case 2:

MoveWindow(hWnd, 0, parentSize.height - mySize.height,

mySize.width, mySize.height, TRUE);

break;

case 3:

MoveWindow(hWnd, 0, 0,

mySize.width, mySize.height, TRUE);

break;

}

status = (status + 1) % 4;

};

// Цикл обработки сообщений

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

{

DispatchMessage(&msg); // Посылает сообщение функции WndProc()

}

return 0;

}

// Оконная функция вызывается операционной системой

// и получает сообщения из очереди для данного приложения

LRESULT CALLBACK WndProc(HWND hWnd, UINT msg,

WPARAM wParam, LPARAM lParam)

{ // Обработчик сообщений

switch(msg)

{

case WM\_RBUTTONDOWN:

handlers[hWnd](hWnd, msg, wParam, lParam);

break;

case WM\_DESTROY : PostQuitMessage(0);

break; // Завершение программы

// Обработка сообщения по умолчанию

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

}

return 0;

}

**Скриншоты**







