

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ  
Кафедра информатики

Индивидуальная практическая работа № 1  
по дисциплине «Системное программирование»

Вариант 6

Факультет: ИНО  
Специальность: ИиТП  
Студент: Дубейковский А.А.  
Группа № 893551  
Зачётная книжка № 75350046

**Условие**

Программа со сложным фоном главного окна: окно содержит изображение (сформированное из примитивов или в виде загружаемого растра). Необходимо предусмотреть корректную перерисовку изображения, а также его отсечение при изменении размеров окна.

В общем, не сразу понял как интерпретировать условие, то ли сделать константную фотографию фоном и чтобы окно могло менять размеры, а фото было всегда его частью одного размера (собственно и это сделал, но потом подумал, что условие наверное о другом).

В итоге пришёл к тому, что требуется сделать открытие окна, где фоном является фото, которое всегда растянуто на макс размер окна, и растягивается/сжимается вместе с соответствующими изменениями окна.

**Код с пояснениями**

**Весь код и видео использования программы будет на гит-хабе (видео много весит, чтобы передавать в СЭО):**

**[https://github.com/ElephantT/System\\_Programming](https://github.com/ElephantT/System_Programming)**

Весь свой код закомментировал (помимо стандартных комментариев созданных Visual Studio), так что наверное нечего будет добавить

```
#include <windows.h>
#include <objidl.h>
#include "framework.h"
#include "sp_ipr1.h"
#include <gdiplus.h>

using namespace Gdiplus;
#pragma comment (lib, "Gdiplus.lib")

#define MAX_LOADSTRING 64

// Глобальные переменные:
HINSTANCE hInst; // текущий экземпляр
WCHAR szTitle[MAX_LOADSTRING]; // Текст строки заголовка
WCHAR szWindowClass[MAX_LOADSTRING]; // имя класса главного окна

// 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)
{
    ULONG_PTR token;
    GdiplusStartupInput input = { 0 };
    input.GdiplusVersion = 1;
    GdiplusStartup(&token, &input, NULL);

    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

    // Инициализация глобальных строк
    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_SPIPR1, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // Выполнить инициализацию приложения:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }

    HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_SPIPR1));

```

```

MSG msg;

// Цикл основного сообщения:
while (GetMessage(&msg, nullptr, 0, 0))
{
    if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}

GdiplusShutdown(token);
return (int) msg.wParam;
}

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_SPIPR1));
    wcex.hCursor        = LoadCursor(nullptr, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName    = MAKEINTRESOURCEW(IDC_SPIPR1);
    wcex.lpszClassName  = szWindowClass;
    wcex.hIconSm        = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI_SMALL));

    return RegisterClassExW(&wcex);
}

```

```
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;
}
```

```

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_ERASEBKGDND: {
            // убираем предыдущий бэкграунд
            RECT rect;
            GetClientRect(hWnd, &rect);
            FillRect(GetDC(hWnd), &rect, (HBRUSH)GetStockObject(BLACK_BRUSH));
            return(1);
        }
        case WM_PAINT: {
            PAINTSTRUCT ps;
            HDC hdc = BeginPaint(hWnd, &ps);

            FillRect(hdc, &ps.rcPaint, (HBRUSH)(COLOR_WINDOW + 1));
            Graphics graphics(hdc);
            // загружаем нашу фотку
            Image image(L"C:/Users/Alek/Documents/University/SP/bsuir_photo.jpg");

            // заносим размеры окна на данный момент в RECT
            RECT rect;
            GetClientRect(hWnd, &rect);

            // переводим RECT в Rect чтобы можно было вызвать graphics.DrawImage
            Rect windiwSize((long)rect.left, (long)rect.top, (long)rect.right, (long)rect.bottom);
            graphics.DrawImage(&image, windiwSize);

            EndPaint(hWnd, &ps);
            return(0);
        }
        case WM_COMMAND:
        {
            int wmId = LOWORD(wParam);
            // Разобрать выбор в меню:
            switch (wmId)
            {
                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_DESTROY:
            PostQuitMessage(0);
            break;
        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
    }
}

```

```

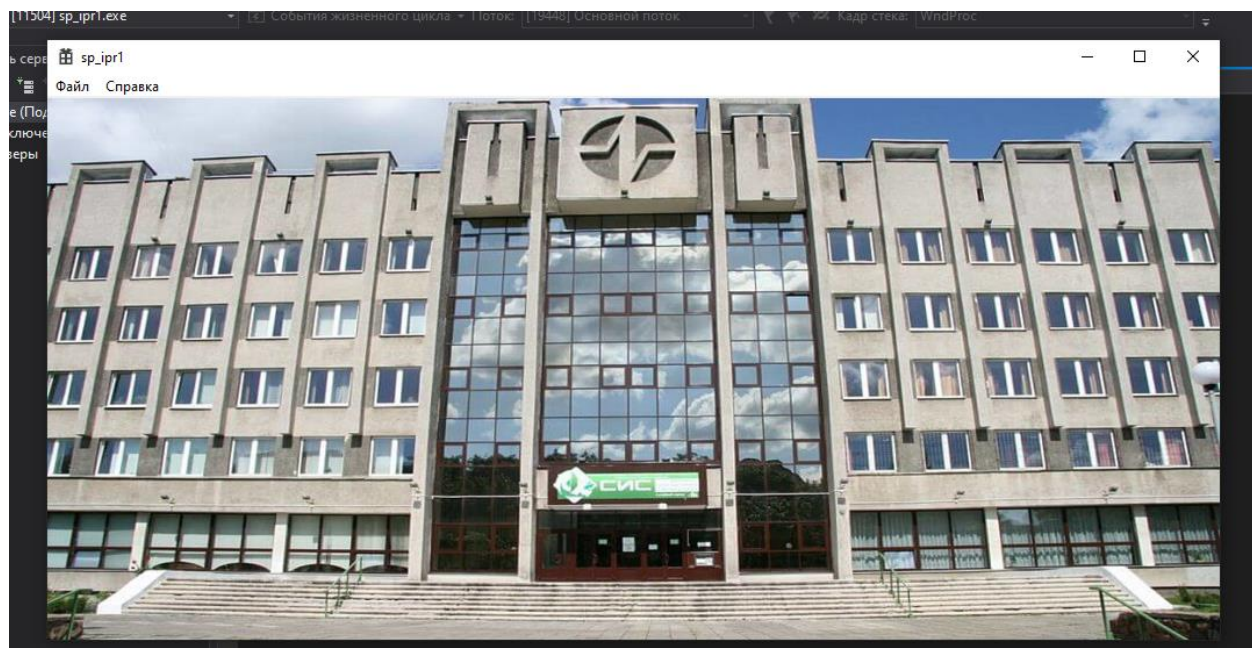
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

// 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;
}

```

Результат:





## Вывод

В ходе написания данной работы получил знания о работе с окнами, бэкграундами, прорисовкой различных элементов в окнах. Так же получил дополнительные знания об обработке различных сообщений, таких как WM\_ERASEBKGD, WM\_PAINT и т.д.