|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1** | |
| **по дисциплине** | |
| «Системное программное обеспечение»  по теме: «Простейшее приложение на базе WIN32/64 API» | |
|  | |
| Выполнил студент группы ИКБО-16-18 | Пономарев А.Р. |
| Принял преподаватель | Волков М.Ю. |

|  |  |  |
| --- | --- | --- |
| Работа выполнена | «\_\_»\_\_\_\_\_\_\_2021 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2021 г. | *(подпись руководителя)* |

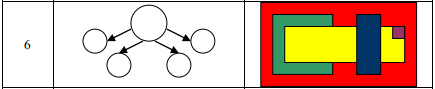
Москва 2021

**Цель работы**

Целью практической работы является изучение основ создания и управления окнами Windows-приложений на базе WIN32/64 API.

**Задание**

Разработать приложение, в котором регистрируются оконные классы, создаются окна классов в соответствии с вариантом задания.



**Ход работы**

1. Cоздать классическое приложение Windows.
2. Зарегистрировать дочерние окна
3. Создать функцию для контроля создаваемых окон.
4. Инициализировать положение окон

Итоговая программа представлена в листинге 1. Результат работы на рисунке 1.

Листинг 1

#include <Windows.h>

#include <tchar.h>

#include <stdlib.h>

#include <string.h>

static TCHAR szWindowClass[] = \_T("DesktopApp");

static TCHAR szTitle[] = \_T("Windows Desktop Guided Tour Application");

HINSTANCE hInst;

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

int CALLBACK WinMain(

\_In\_ HINSTANCE hInstance,

\_In\_opt\_ HINSTANCE hPrevInstane,

\_In\_ LPSTR lpCmdLine,

\_In\_ int nCmdShow)

{

//window 1

WNDCLASSEX wcex1;

HBRUSH redBrush = CreateSolidBrush(RGB(255, 0, 0));

ZeroMemory(&wcex1, sizeof(WNDCLASSEX));

wcex1.cbSize = sizeof(WNDCLASSEX);

wcex1.style = CS\_HREDRAW | CS\_VREDRAW;

wcex1.lpfnWndProc = WndProc;

wcex1.cbClsExtra = 0;

wcex1.cbWndExtra = 0;

wcex1.hInstance = hInstance;

wcex1.hIcon = LoadIcon(hInstance, IDI\_APPLICATION);

wcex1.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex1.hbrBackground = redBrush;

wcex1.lpszMenuName = NULL;

wcex1.lpszClassName = L"window class1";

wcex1.hIconSm = LoadIcon(wcex1.hInstance, IDI\_APPLICATION);

if (!RegisterClassEx(&wcex1)) {

MessageBox(NULL,

\_T("Window class creation failed for window 1"),

\_T("Window Class Failed"),

MB\_ICONERROR);

return 1;

}

HWND hWnd1 = CreateWindow(

wcex1.lpszClassName,

L"Parent Window",

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, CW\_USEDEFAULT,

1400, 800,

NULL,

NULL,

hInstance,

NULL

);

if (!hWnd1)

{

MessageBox(NULL,

\_T("Call to CreateWindow1 failed!"),

\_T("Window Creation Filed"),

MB\_ICONERROR);

return 1;

}

ShowWindow(hWnd1,

nCmdShow);

UpdateWindow(hWnd1);

//window 2

WNDCLASSEX wcex2;

HBRUSH greenBrush = CreateSolidBrush(RGB(51, 153, 102));

ZeroMemory(&wcex2, sizeof(WNDCLASSEX));

wcex2.cbSize = sizeof(WNDCLASSEX);

wcex2.style = CS\_HREDRAW | CS\_VREDRAW;

wcex2.lpfnWndProc = WndProc;

wcex2.cbClsExtra = 0;

wcex2.cbWndExtra = 0;

wcex2.hInstance = hInstance;

wcex2.hIcon = LoadIcon(hInstance, IDI\_APPLICATION);

wcex2.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex2.hbrBackground = greenBrush;

wcex2.lpszMenuName = NULL;

wcex2.lpszClassName = L"window class 2";

wcex2.hIconSm = LoadIcon(wcex2.hInstance, IDI\_APPLICATION);

if (!RegisterClassEx(&wcex2)) {

MessageBox(NULL,

\_T("Call to RegisterClassEx failed!"),

\_T("Window Class Failed!"),

NULL);

return 1;

}

HWND hWnd2 = CreateWindow(

wcex2.lpszClassName,

L"Child window",

WS\_OVERLAPPEDWINDOW,

100, 100,

600, 600,

NULL,

NULL,

hInstance,

NULL

);

if (!hWnd2)

{

MessageBox(NULL,

\_T("Call to CreateWindow 2 failed!"),

\_T("Window call filed"),

NULL);

return 1;

}

ShowWindow(hWnd2,

nCmdShow);

UpdateWindow(hWnd2);

//window 3

WNDCLASSEX wcex3;

HBRUSH yellowBrush = CreateSolidBrush(RGB(255, 255, 0));

ZeroMemory(&wcex3, sizeof(WNDCLASSEX));

wcex3.cbSize = sizeof(WNDCLASSEX);

wcex3.style = CS\_HREDRAW | CS\_VREDRAW;

wcex3.lpfnWndProc = WndProc;

wcex3.cbClsExtra = 0;

wcex3.cbWndExtra = 0;

wcex3.hInstance = hInstance;

wcex3.hIcon = LoadIcon(hInstance, IDI\_APPLICATION);

wcex3.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex3.hbrBackground = yellowBrush;

wcex3.lpszMenuName = NULL;

wcex3.lpszClassName = L"window class 3";

wcex3.hIconSm = LoadIcon(wcex3.hInstance, IDI\_APPLICATION);

if (!RegisterClassEx(&wcex3)) {

MessageBox(NULL,

\_T("Call to RegisterClassEx failed!"),

\_T("Window Class Failed!"),

NULL);

return 1;

}

HWND hWnd3 = CreateWindow(

wcex3.lpszClassName,

L"Child window 2",

WS\_OVERLAPPEDWINDOW,

250, 250,

1100, 300,

NULL,

NULL,

hInstance,

NULL

);

if (!hWnd3)

{

MessageBox(NULL,

\_T("Call to CreateWindow 3 failed!"),

\_T("Window call filed"),

NULL);

return 1;

}

ShowWindow(hWnd3,

nCmdShow);

UpdateWindow(hWnd3);

//window 4

WNDCLASSEX wcex4;

HBRUSH purpleBrush = CreateSolidBrush(RGB(153, 51, 102));

ZeroMemory(&wcex4, sizeof(WNDCLASSEX));

wcex4.cbSize = sizeof(WNDCLASSEX);

wcex4.style = CS\_HREDRAW | CS\_VREDRAW;

wcex4.lpfnWndProc = WndProc;

wcex4.cbClsExtra = 0;

wcex4.cbWndExtra = 0;

wcex4.hInstance = hInstance;

wcex4.hIcon = LoadIcon(hInstance, IDI\_APPLICATION);

wcex4.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex4.hbrBackground = purpleBrush;

wcex4.lpszMenuName = NULL;

wcex4.lpszClassName = L"window class 4";

wcex4.hIconSm = LoadIcon(wcex4.hInstance, IDI\_APPLICATION);

if (!RegisterClassEx(&wcex4)) {

MessageBox(NULL,

\_T("Call to RegisterClassEx failed!"),

\_T("Window Class Failed!"),

NULL);

return 1;

}

HWND hWnd4 = CreateWindow(

wcex4.lpszClassName,

L"Child window 3",

WS\_OVERLAPPEDWINDOW,

1200, 280,

150, 150,

NULL,

NULL,

hInstance,

NULL

);

if (!hWnd4)

{

MessageBox(NULL,

\_T("Call to CreateWindow 4 failed!"),

\_T("Window call filed"),

NULL);

return 1;

}

ShowWindow(hWnd4,

nCmdShow);

UpdateWindow(hWnd4);

//window 4

WNDCLASSEX wcex5;

HBRUSH blueBrush = CreateSolidBrush(RGB(0, 51, 102));

ZeroMemory(&wcex5, sizeof(WNDCLASSEX));

wcex5.cbSize = sizeof(WNDCLASSEX);

wcex5.style = CS\_HREDRAW | CS\_VREDRAW;

wcex5.lpfnWndProc = WndProc;

wcex5.cbClsExtra = 0;

wcex5.cbWndExtra = 0;

wcex5.hInstance = hInstance;

wcex5.hIcon = LoadIcon(hInstance, IDI\_APPLICATION);

wcex5.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex5.hbrBackground = blueBrush;

wcex5.lpszMenuName = NULL;

wcex5.lpszClassName = L"window class 5";

wcex5.hIconSm = LoadIcon(wcex5.hInstance, IDI\_APPLICATION);

if (!RegisterClassEx(&wcex5)) {

MessageBox(NULL,

\_T("Call to RegisterClassEx failed!"),

\_T("Window Class Failed!"),

NULL);

return 1;

}

HWND hWnd5 = CreateWindow(

wcex5.lpszClassName,

L"Child window 4",

WS\_OVERLAPPEDWINDOW,

800, 100,

150, 600,

NULL,

NULL,

hInstance,

NULL

);

if (!hWnd5)

{

MessageBox(NULL,

\_T("Call to CreateWindow 5 failed!"),

\_T("Window call filed"),

NULL);

return 1;

}

ShowWindow(hWnd5,

nCmdShow);

UpdateWindow(hWnd5);

SetParent(hWnd2, hWnd1);

SetParent(hWnd3, hWnd1);

SetParent(hWnd4, hWnd1);

SetParent(hWnd5, hWnd1);

MSG msg;

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

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return (int)msg.wParam;

};

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

{

PAINTSTRUCT ps;

HDC hdc;

switch (message)

{

case WM\_PAINT:

hdc = BeginPaint(hWnd, &ps);

TextOut(hdc,

5, 5,

NULL, NULL);

EndPaint(hWnd, &ps);

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

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

break;

}

return 0;

};

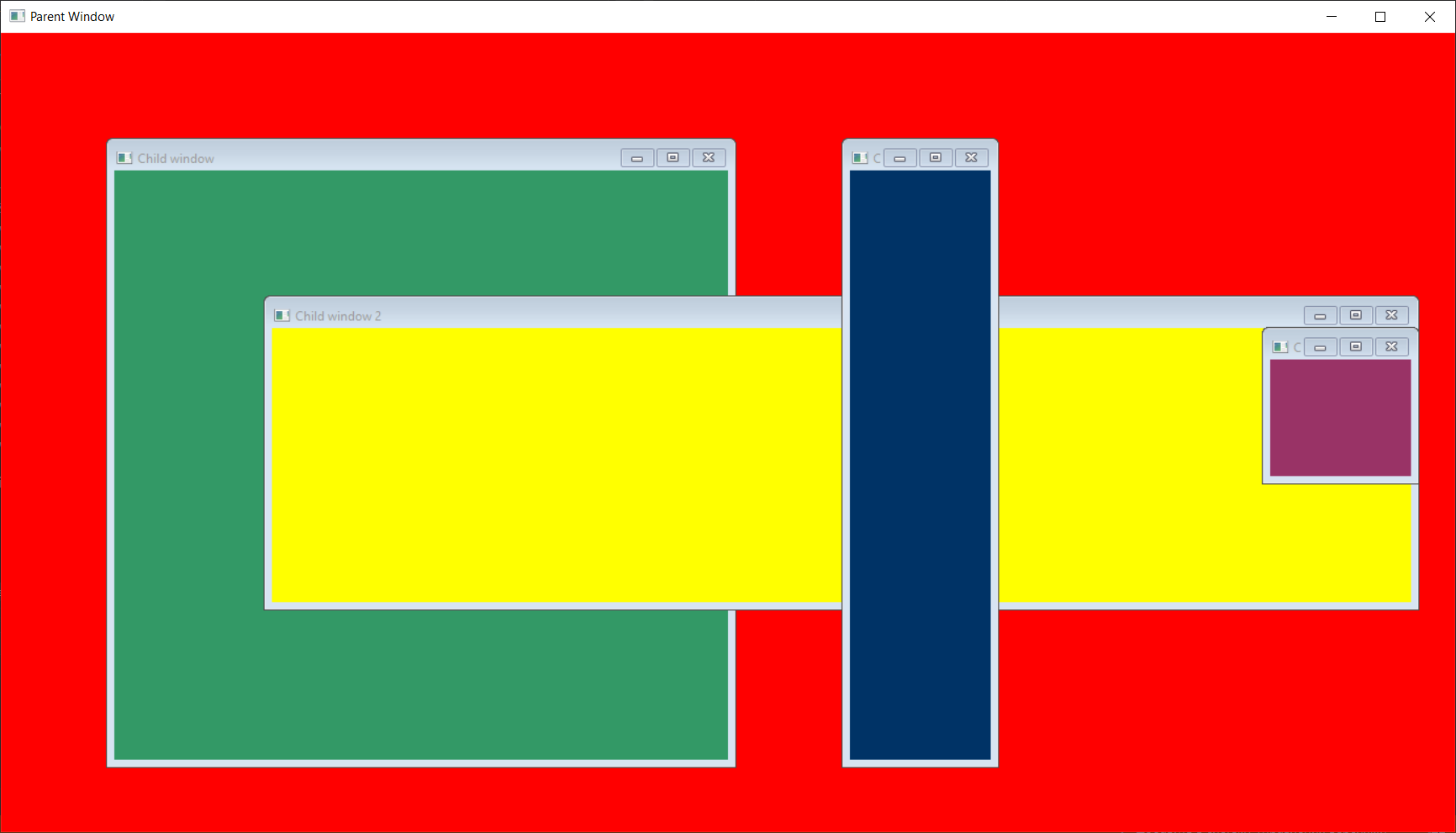


Рисунок 1. Оконное приложение

**Вывод**

В ходе практической работы был проведён ряд операций по формированию классических форм интерфейса. Освоены навыки работы с документацией и использованием графических возможностей WINAPI.

**Список использованных источников**

1. Пошаговое руководство. Создание классического оконного приложение [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/cpp/windows/walkthrough-creating-windows-desktop-applications-cpp?view=msvc-160
2. Администрирование в Windows 10. [Электронный ресурс] : [https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10 /](https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10%20/) Ричард Ворд 2020 — 100 с.
3. «Тайм-менеджмент для системных администраторов». / Лимончелли Т.А. М.: Символ-плюс 2007 — 247 с. ISBN:978-5-93286-090-8