**4 ЛАБОРАТОРНАЯ РАБОТА №4**

**ОТОБРАЖЕНИЕ ОКОН С ИСПОЛЬЗОВАНИЕМ РЕГИОНОВ.**

**БИБЛИОТЕКА WINMM**

**Цель работы:** освоить понятие региона и способы задания регионов окна. Получить практические навыки по изменению формы окна при помощи регионов. Изучить способы отображения текста поверх всех окон. Получить навыки использования мультимедийной библиотеки Windows.

**4.1 Листинг программы, полученной в результате выполнения пункта 4.3**

#include <windows.h>

#include <windowsx.h>

WNDCLASS hWc;

HWND hWnd;

LPCTSTR WCName = L"MyClass";

MSG Msg;

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

HDC hdc, hdcScreen;

HRGN hWindowRgn, hHoleRgn, hNewWindowRgn;

RECT Rt;

switch (Msg){

// в обработчике события WM\_SIZE

case WM\_SIZE:

// возвращает координаты окна

GetWindowRect(hWnd, &Rt);

// так как необходимо задавать регион окна относительно

// его левого верхнего угла, нужно его сместить так, чтобы

// верхний левый угол прямоугольника был (0, 0)

Rt.right = Rt.right - Rt.left;

Rt.bottom = Rt.bottom - Rt.top;

Rt.left = Rt.top = 0;

if (Rt.right < Rt.bottom){

Rt.bottom = Rt.right;

} else {

Rt.right = Rt.bottom;

}

// создание региона в виде эллипса, ограниченного прямоугольником

// Rt

hWindowRgn = CreateEllipticRgnIndirect(&Rt);

// создание маленького эллипса в центре

hHoleRgn = CreateEllipticRgn(

Rt.left + (Rt.right - Rt.left) / 4,

Rt.top + (Rt.bottom - Rt.top) / 4,

Rt.right - (Rt.right - Rt.left) / 4,

Rt.bottom - (Rt.bottom - Rt.top) / 4);

hNewWindowRgn = CreateRectRgn(0, 0, 0, 0);

// объединение полученных регионов

CombineRgn(hNewWindowRgn, hWindowRgn, hHoleRgn, RGN\_XOR);

// установка созданного региона для окна с перерисовкой

// этого окна

SetWindowRgn(hWnd, hNewWindowRgn, TRUE);

// удаление ранее созданного региона

DeleteObject(hWindowRgn);

// удаление ранее созданного региона

DeleteObject(hHoleRgn);

// удаление ранее созданного региона

DeleteObject(hNewWindowRgn);

return 0;

case WM\_DESTROY:

PostMessage(hWnd, WM\_QUIT, 0, 0);

break;

default:

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

};

};

int WINAPI WinMain(HINSTANCE hInst, HINSTANCE hPrevInst, LPSTR lpCmdLine, int nCmdShow){

hWc.cbClsExtra = 0;

hWc.cbWndExtra = 0;

hWc.hbrBackground = (HBRUSH)COLOR\_APPWORKSPACE;

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

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

hWc.hInstance = hInst;

hWc.lpfnWndProc = WndProc;

hWc.lpszClassName = WCName;

hWc.lpszMenuName = NULL;

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

RegisterClass(&hWc);

hWnd = CreateWindow(WCName, L"List Processes", WS\_OVERLAPPEDWINDOW,

50, 50, 400, 300, NULL, NULL, hInst, 0);

ShowWindow(hWnd, SW\_SHOW);

while (GetMessage(&Msg, NULL, NULL, NULL)){

DispatchMessage(&Msg);

};

return 0;

};

**4.2 Листинг программы заданной на самостоятельную работу**

#include <windows.h>

#include <windowsx.h>

WNDCLASS hWc;

HWND hWnd;

LPCTSTR WCName = L"MyClass";

MSG Msg;

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

HDC hdc, hdcScreen;

HRGN hWindowRgn, hHoleRgn, hNewWindowRgn;

RECT Rt;

HWND hWndText = CreateWindowEx(WS\_EX\_TOPMOST | WS\_EX\_TRANSPARENT | WS\_EX\_LAYERED,

L"message", L"", WS\_POPUP,

0, 0, 500, 400, NULL, NULL, NULL, NULL);

SetLayeredWindowAttributes(hWndText, RGB(0,0,0), 0, LWA\_COLORKEY);

switch (Msg){

case WM\_LBUTTONDOWN:{

GetWindowRect(hWnd, &Rt);

POINT vertices[3];

vertices[0].x = (Rt.right - Rt.left)/2;

vertices[0].y = 0;

vertices[1].x = 0;

vertices[1].y = Rt.bottom - Rt.top;

vertices[2].x = Rt.right - Rt.left;

vertices[2].y = vertices[1].y;

hWindowRgn = CreatePolygonRgn(vertices, 3, ALTERNATE);

SetWindowRgn(hWnd, hWindowRgn, TRUE);

DeleteObject(hWindowRgn);

return 0;

}

case WM\_RBUTTONDOWN:{

ShowWindow(hWndText, SW\_NORMAL);

UpdateWindow(hWndText);

HDC dc = GetWindowDC(hWndText);

SelectObject(dc, GetStockObject(BLACK\_BRUSH));

PatBlt(dc, 0, 0, 500, 100, PATCOPY);

TCHAR str[] = L"TEST TEST TEST";

TextOut(dc, 200, 100, str, wcslen(str));

ReleaseDC(hWndText, dc);

return 1;

}

case WM\_DESTROY:

PostMessage(hWnd, WM\_QUIT, 0, 0);

break;

default:

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

};

};

**Выводы**

В ходе лабораторной работы было изучено понятие региона и способы задания регионов окна. Было создано приложение, которое меняет форму окна на треугольную, а также выводит текст, который отображается поверх всех окон.