#include <windows.h>

//#include <unistd.h>

#include <stdbool.h>

//#include < stdio.h >

#include <dos.h>

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

void printHat(HDC hdc, HPEN hPen, POINT pol[5]) {

HBRUSH sBrush;

sBrush = CreateSolidBrush(RGB(255, 98, 0));

SelectObject(hdc, sBrush);

sBrush = CreateSolidBrush(RGB(168, 5, 24));

SelectObject(hdc, sBrush);

Polygon(hdc, pol, 5);

DeleteObject(sBrush);

DeleteObject(hPen);

}

void printLeftHand(HDC hdc, POINT p) {

MoveToEx(hdc, 300, 185, NULL); //сделать текущими координаты x1, y1

LineTo(hdc, p.x, p.y); //230 150

}

void printRightHand(HDC hdc, POINT p) {

MoveToEx(hdc, 410, 185, NULL); //сделать текущими координаты x1, y1

LineTo(hdc, p.x, p.y); //480, 230

}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR szCmdLine, int iCmdShow)

//iCmdshow 1-выводится окно

{

static char szAppName[] = "HelloWin";//имя класса окна

HWND hwnd;//структура описателя окна

MSG msg;//Структура собщения

WNDCLASSEX wndclass;//Структура класса окна

wndclass.cbSize = sizeof(wndclass);// Длина структуры

wndclass.style = CS\_HREDRAW | CS\_VREDRAW;//Идентиф., указ., что все окна перерисовыв при изм гор и верт размеров

wndclass.lpfnWndProc = WndProc;//Адрес оконной процедуры, исп для всех окон класса

wndclass.cbClsExtra = 0;//Резервир доп память

wndclass.cbWndExtra = 0;//Резервир доп память

wndclass.hInstance = hInstance;//описатель экземпляра программы

wndclass.hIcon = LoadIcon(NULL, IDI\_APPLICATION);//Значок слева вверху-стандартный

wndclass.hCursor = LoadCursor(NULL, IDC\_ARROW);//стандартный курсор

wndclass.hbrBackground = (HBRUSH)GetStockObject(WHITE\_BRUSH);//Заливка белым цветом фона

wndclass.lpszMenuName = NULL;//Меню класса окна

wndclass.lpszClassName = szAppName;// Имя класса окна

wndclass.hIconSm = LoadIcon(NULL, IDI\_APPLICATION);

RegisterClassEx(&wndclass);//Ф-ия, регистрирующая класс окна

//CreateWindow возвращает описатель созданного окна

hwnd = CreateWindow(szAppName, //Имя класса окна

"Лабораторная по PW", // Заголовок окна

WS\_OVERLAPPEDWINDOW, // Стиль окна - стандартный с кнопками сворачивания, рамкой..

CW\_USEDEFAULT, // Начальное положение по х

CW\_USEDEFAULT, // Начальное положение по у

CW\_USEDEFAULT, // Начальный размер по х

CW\_USEDEFAULT, // Начальный размер по у

NULL, // Описатель родительского окна

NULL, // Описатель меню окна

hInstance, // Описатель экземпляра программы

NULL); // Параметры создания

ShowWindow(hwnd, iCmdShow);//Показ окна

UpdateWindow(hwnd);//Перерисовка рабочей области

while (GetMessage(&msg, NULL, 0, 0))//извлекает сообщ из очереди сообщ(увеличили окно, закрыли и тд) 0-обраб все сообщ

{

TranslateMessage(&msg);//Передаёт структуру в Windows

DispatchMessage(&msg);//Передаёт структуру в Windows

}

return msg.wParam;

}

LRESULT CALLBACK WndProc(HWND hwnd,//Описатель получающего сообщение окна

UINT iMsg, //Число, идентифицирующее сообщение

WPARAM wParam, //Параметры сообщения

LPARAM lParam) //Параметры сообщения

{

HDC hdc;//описатель контекста диспозитива, зоны клиента

PAINTSTRUCT ps;//Структура для рисования

RECT rect;//прямоугольник

static bool hat;

switch (iMsg)

{

case WM\_CREATE:

SetTimer(hwnd, 1, 1000, 0);

hat = false;

break;

case WM\_TIMER:

//InvalidateRect(hwnd, NULL, TRUE);

UpdateWindow(hwnd);

hdc = GetDC(hwnd);

//InvalidateRect(hwnd, NULL, TRUE);

HPEN hPen;

hPen = CreatePen(PS\_SOLID, 5, RGB(66, 39, 22)); //Создаётся объект

SelectObject(hdc, hPen);

POINT leftDown;

leftDown.x = 230;

leftDown.y = 150;

POINT rightUp;

rightUp.x = 480;

rightUp.y = 230;

printLeftHand(hdc, leftDown);

printRightHand(hdc, rightUp);

if (hat) {

for (int i = 0; i < 10; i+=1) {

InvalidateRect(hwnd, NULL, TRUE);

UpdateWindow(hwnd);

POINT pol[5];

pol[0].x = 330\*i;

pol[0].y = 80 \* i;

pol[1].x = 350\*i;

pol[1].y = 40 \* i;

pol[2].x = 370\*i;

pol[2].y = 40 \* i;

pol[3].x = 390\*i;

pol[3].y = 80 \* i;

pol[4].x = 330\*i;

pol[4].y = 80 \* i;

printHat(hdc, hPen, pol);

Sleep(3);

}

/\*HBRUSH fBrush;

InvalidateRect(hwnd, NULL, TRUE);

UpdateWindow(hwnd);

fBrush = CreateSolidBrush(RGB(6, 26, 77));

SelectObject(hdc, fBrush);

for (int i = 0; i < 5; i += 0.5) {

int a = 350 \* i, b = 160 \* i, c = 365 \* i, d = 175 \* i;

Ellipse(hdc, a, b, c, d);

Sleep(3);

}\*/

}

else {

for (int i = 0; i < 10; i += 1) {

InvalidateRect(hwnd, NULL, TRUE);

UpdateWindow(hwnd);

POINT pol[5];

pol[0].x = 330 \* (-i);

pol[0].y = 80 \*(- i);

pol[1].x = 350 \* (-i);

pol[1].y = 40 \* (-i);

pol[2].x = 370 \* (-i);

pol[2].y = 40 \* (-i);

pol[3].x = 390 \* (-i);

pol[3].y = 80 \* (-i);

pol[4].x = 330 \* (-i);

pol[4].y = 80 \* (-i);

printHat(hdc, hPen, pol);

Sleep(3);

}

/\*

HBRUSH fBrush;

InvalidateRect(hwnd, NULL, TRUE);

UpdateWindow(hwnd);

fBrush = CreateSolidBrush(RGB(6, 26, 77));

SelectObject(hdc, fBrush);

for (int i = 0; i < 5; i += 0.5) {

int a = 350 \* i, b = 160 \* i, c = 365 \* i, d = 175 \* i;

Ellipse(hdc, a, b, c, d);

Sleep(3);

}\*/

/\*if (hat){

HBRUSH sBrush;

sBrush = CreateSolidBrush(RGB(6, 26, 77));

SelectObject(hdc, sBrush);

for (int i = 0; i < 5; i += 0.5) {

int a = 350 \* i, b = 160 \* i, c = 365 \* i, d = 175 \* i;

Ellipse(hdc, a, b, c, d);

Sleep(3);

}

}

else {

HBRUSH sBrush;

sBrush = CreateSolidBrush(RGB(6, 26, 77));

SelectObject(hdc, sBrush);

for (int i = 0; i < 5; i += 0.5) {

int a = 350 \* i, b = 160 \* i, c = 365 \* i, d = 175 \* i;

Ellipse(hdc, a, b, c, d);

Sleep(3);

}\*/

}

ReleaseDC(hwnd, hdc);

hat = !hat;

break;

case WM\_PAINT:

hdc = BeginPaint(hwnd, &ps);

//здесь можно вставить функции рисования

HBRUSH sBrush;

sBrush = CreateSolidBrush(RGB(107, 178, 209));

SelectObject(hdc, sBrush);

//Тело

Ellipse(hdc, 330, 80, 390, 140); //эллипс будет закрашен

Arc(hdc, 330, 70, 390, 130, 340, 120, 380, 120);

Ellipse(hdc, 300, 140, 410, 250); //эллипс будет закрашен

Ellipse(hdc, 250, 250, 450, 450); //эллипс будет закрашен

sBrush = CreateSolidBrush(RGB(28, 97, 28));

SelectObject(hdc, sBrush);

//Глаза

Ellipse(hdc, 340, 90, 350, 100);

Ellipse(hdc, 370, 90, 380, 100);

POINT poly[4];

poly[0].x = 360;

poly[0].y = 105;

poly[1].x = 380;

poly[1].y = 113;

poly[2].x = 360;

poly[2].y = 120;

poly[3].x = 360;

poly[3].y = 105;

sBrush = CreateSolidBrush(RGB(255, 98, 0));

SelectObject(hdc, sBrush);

Polygon(hdc, poly, 4);

sBrush = CreateSolidBrush(RGB(6,26,77));

SelectObject(hdc, sBrush);//Пуговицы

Ellipse(hdc, 350, 190, 365, 205);

//Ellipse(hdc, 350, 160, 365, 175);

Ellipse(hdc, 350, 220, 365, 235);

/\*HPEN hPen; //Объявляется кисть

hPen = CreatePen(PS\_SOLID, 5, RGB(66, 39, 22)); //Создаётся объект

SelectObject(hdc, hPen); //Объект делается текущим\*/

/\*MoveToEx(hdc, 300, 185, NULL); //сделать текущими координаты x1, y1

LineTo(hdc, 230, 150);

MoveToEx(hdc, 410, 185, NULL); //сделать текущими координаты x1, y1

LineTo(hdc, 480, 150);\*/

//обновляем окно

ValidateRect(hwnd, NULL);

//заканчиваем рисовать

EndPaint(hwnd, &ps);

break;

DeleteObject(sBrush);

//DeleteObject(hPen);

case WM\_DESTROY://Заканчивает программу

KillTimer(hwnd, 1);

PostQuitMessage(0);

return 0;

}

return DefWindowProc(hwnd, iMsg, wParam, lParam);//Все сообщения, не обрабатываемые оконной процедурой, возвр значения сообщ

}