

Лекция 3. Графический интерфейс в Windows API

Архитектура ОС Windows

16 октября 2013 г.

Вывод приложением графической информации



Рис. 1: структура взаимодействия приложения с устройством вывода

Атрибуты контекста отображения

Атрибут	Функция установки
Цвет фона	SetBkColor()
Режим фона	SetBkMode()
Режим рисования	SetROP2()
Цвет текста	SetTextColor()
Шрифт	SelectObject()
Расстояние между символами	SetTextCharactersExtra()
Цветовая палитра	SelectPalette()
Кисть	SelectObject()
Начальные координаты кисти	SetBrushOrgEx()
Метод закрашивания многоугольников	SetPolyFillMode()
Перо	SelectObject()
Битовое изображение	SelectObject()
Текущая позиция пера	MoveToEx()

Таблица 1: основные атрибуты контекста отображения

Атрибуты контекста отображения (окончание)

Атрибут	Функция установки
Режим растяжения	SetStretchBltMode()
Область ограничения изображения	SelectClipRgn()
Режим отображения	SetMapMode()
Начало системы координат окна	SetWindowOrgEx()
Начало системы физических координат	SetViewportOrgEx()
Масштаб осей окна	SetWindowExtEx()
Масштаб осей физических координат	SetViewportExtEx()

Таблица 2: основные атрибуты контекста отображения

Работа с контекстом отображения

Порядок работы

- 1 Получение или создание контекста отображения.
- 2 Установка необходимых атрибутов в контексте отображения.
- 3 Выполнение операций рисования.
- 4 Освобождение или удаление контекста отображения.

Виды контекстов отображения

Название	Англ.	Функция получения
Общий контекст отображения	common display context	BeginPaint()/GetDC()
Контекст отображения окна	window display context	GetWindowDC()
Контекст физического устройства	device context	CreateDC()/GetDC()
Информационный контекст	information context	CreateIC()
Контекст для памяти	Memory context	CreateComptibleDC()
Контекст для метафайла	Metafile context	CreateMetaFile()

Таблица 3: основные виды контекстов отображения

Пример обработки сообщения перерисовки

Пример

```
switch (uMessage)
{
    case WM_PAINT:
    {
        // Получение контекста устройства
        PAINTSTRUCT paint_s;
        HDC hDC = BeginPaint(hWnd, &paint_s);
        // Вывод строки
        TextOut(hDC, 10, 20, TEXT("Сообщение WM_PAINT"), 18);
        // Возврат контекста устройства
        EndPaint(hWnd, &paint_s);
        break;
    }
}
```

Функция TextOut()

Определение функции TextOut()

```
BOOL TextOut(  
    __in    HDC    hDC,  
    __in    int   nXStart,  
    __in    int   nYStart,  
    __in    LPCTSTR lpctszString,  
    __in    int   nStringCharsNum  
);
```

Пример обработки сообщения мыши

Пример

```
switch (uMessage)
{
    // ...
    case WM_LBUTTONDOWN:
    {
        HDC hDC = GetDC(hWnd);
        TextOut(
            hDC, 10, 40, TEXT("Сообщение WM_LBUTTONDOWN"), 24);
        ReleaseDC(hWnd, hDC);
        break;
    }
    // ...
}
```

Пример

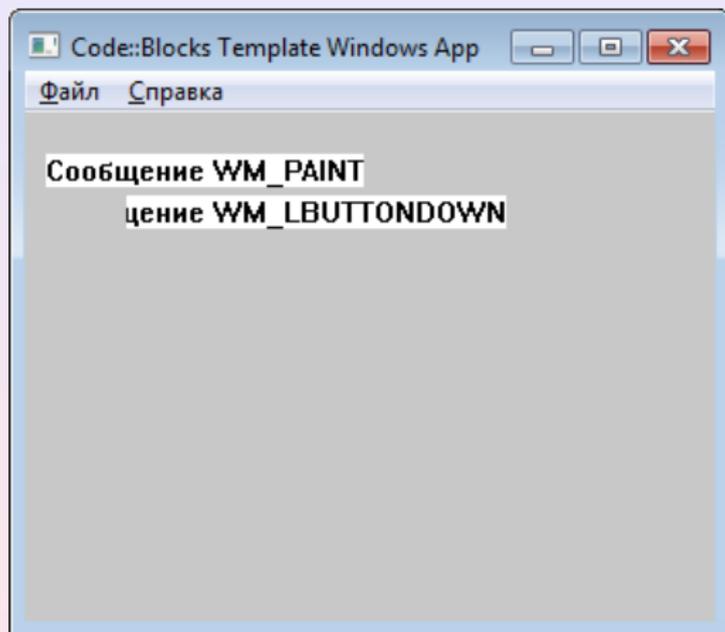


Рис. 2: обработка перерисовки окна

Пример обработки передвижения мыши

Пример

```
case WM_MOUSEMOVE:
{
    if (wParam & MK_LBUTTON && wParam & MK_SHIFT)
    {
        int nX = GET_X_LPARAM(lParam);    // из windowsx.h
        int nY = GET_Y_LPARAM(lParam);
        HDC hDC = GetDC(hWnd);
        SetPixel(hDC, nX, nY, RGB(0, 0, 0));
        ReleaseDC(hWnd, hDC);
    }
    break;
}
```

Функция SetMapMode()

Определение SetMapMode()

```
int SetMapMode(  
    __in     HDC     hDC,  
    __in     int    nMapMode  
);
```

MM_TEXT

MM_ISOTROPIC

MM_ANISOTROPIC

Таблица 4: основные режимы отображения

Функции SetViewportOrgEx() и SetWindowOrgEx()

Определение SetViewportOrgEx()

```
BOOL SetViewportOrgEx(  
    __in     HDC     hDC,  
    __in     int     nX,  
    __in     int     nY,  
    __out    LPPPOINT lpPoint  
);
```

Определение SetWindowOrgEx()

```
BOOL SetWindowOrgEx(  
    __in     HDC     hDC,  
    __in     int     nX,  
    __in     int     nY,  
    __out    LPPPOINT lpPoint  
);
```

Функции SetViewportExtEx() и SetWindowExtEx()

Определение SetViewportExtEx()

```
BOOL SetViewportExtEx(  
    __in    HDC    hDC,  
    __in    int    nXExtent,  
    __in    int    nYExtent,  
    __out   LPSIZE lpSize  
);
```

Определение SetWindowExtEx()

```
BOOL SetWindowExtEx(  
    __in    HDC    hDC,  
    __in    int    nXExtent,  
    __in    int    nYExtent,  
    __out   LPSIZE lpSize  
);
```

Преобразование координат

Логические \rightarrow физические

$$x_v = x_{ov} + \frac{x_{ev}}{x_{ew}} (x_w - x_{ow})$$

$$y_v = y_{ov} + \frac{y_{ev}}{y_{ew}} (y_w - y_{ow})$$

Преобразование координат

Логические \rightarrow физические

$$x_v = x_{ov} + \frac{x_{ev}}{x_{ew}} (x_w - x_{ow})$$
$$y_v = y_{ov} + \frac{y_{ev}}{y_{ew}} (y_w - y_{ow})$$

Физические \rightarrow логические

$$x_w = x_{ow} + \frac{x_{ew}}{x_{ev}} (x_v - x_{ov})$$
$$y_w = y_{ow} + \frac{y_{ew}}{y_{ev}} (y_v - y_{ov})$$

Функция GetClientRect()

Определение GetClientRect()

```
BOOL WINAPI GetClientRect(  
    __in    HWND    hWnd,  
    __out   LPRECT  lpRect  
);
```

Определение RECT

```
typedef struct _RECT {  
    LONG left;  
    LONG top;  
    LONG right;  
    LONG bottom;  
} RECT, *PRECT;
```

Функция SelectObject()

Определение SelectObject()

```
HGDIOBJ WINAPI SelectObject(  
    __in    HDC      hDC,  
    __in    HGDIOBJ hGDIObj  
);
```

Возможные объекты

- битовые изображения (только контекст памяти);
- кисти;
- перья;
- шрифты;
- области изображения.

Функция GetStockObject()

Определение GetStockObject()

```
HGDIOBJ GetStockObject(  
    __in    int    fnObject);
```

Кисти	Перья	Шрифты
BLACK_BRUSH	BLACK_PEN	ANSI_FIXED_FONT
DKGRAY_BRUSH		ANSI_VAR_FONT
GRAY_BRUSH		OEM_FIXED_FONT
LTGRAY_BRUSH		DEVICE_DEFAULT_FONT
WHITE_BRUSH	WHITE_PEN	
DC_BRUSH	DC_PEN	
NULL_BRUSH	NULL_PEN	

Таблица 5: идентификаторы объектов

Функции `Ellipse()` и `Rectangle()`

Определение `Ellipse()`

```
BOOL Ellipse(  
    __in    HDC    hDC,  
    __in    int    nLeftRect,  
    __in    int    nTopRect,  
    __in    int    nRightRect,  
    __in    int    nBottomRect  
);
```

Определение `Rectangle()`

```
BOOL Rectangle(  
    __in    HDC    hDC,  
    __in    int    nLeftRect,  
    __in    int    nTopRect,  
    __in    int    nRightRect,  
    __in    int    nBottomRect  
);
```

Пример центрирования изображения

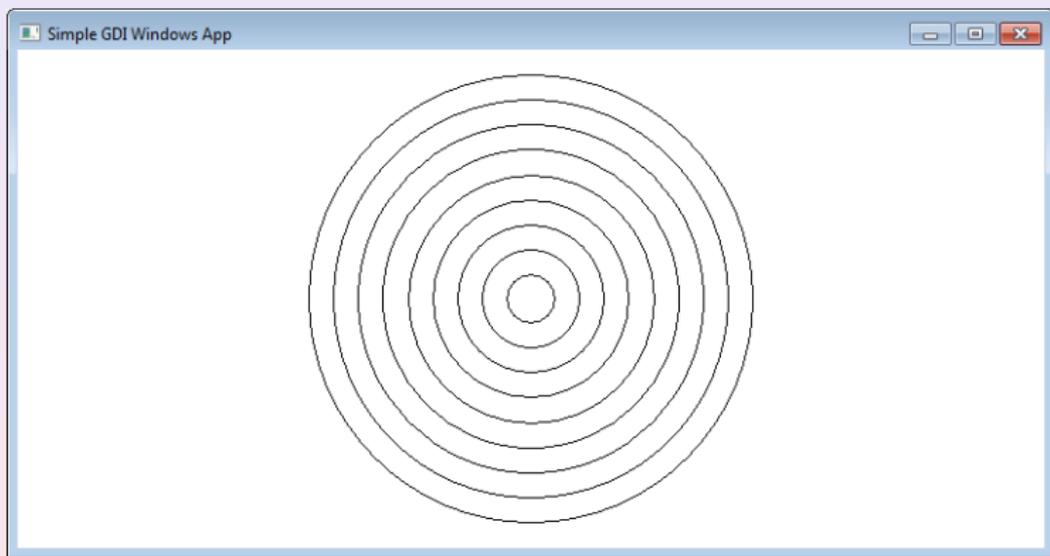


Рис. 3: центрирование изображения в окне

Пример центрирования изображения (продолжение)

Пример

```
int APIENTRY WinMain(  
    HINSTANCE hThisInstance,  
    HINSTANCE hPrevInstance,  
    LPTSTR lpszCmdLine,  
    int nCmdShow)  
{  
    WNDCLASSEX wnd_class;  
    wnd_class.cbSize = sizeof (WNDCLASSEX);  
    wnd_class.style = CS_HREDRAW | CS_VREDRAW;  
    // ...  
    //  
    return messages.wParam;  
}    // WinMain()
```

Пример центрирования изображения (продолжение)

Пример (продолжение)

```
LRESULT CALLBACK WindowProcedure(  
    HWND hWnd, UINT uMessage, WPARAM wParam, LPARAM lParam)  
{  
    switch (uMessage)  
    {  
        case WM_PAINT:  
        {  
            PAINTSTRUCT paint_s;  
            HDC hDC = BeginPaint(hWnd, &paint_s);  
            //
```

Пример центрирования изображения (продолжение)

Пример (продолжение)

```
RECT rect;  
GetClientRect(hWnd, &rect);  
SetMapMode(hDC, MM_ISOTROPIC);  
SetWindowExtEx(hDC, 1000, 1000, NULL);  
SetViewportOrgEx(hDC, rect.right / 2, rect.bottom / 2, NULL);  
SetViewportExtEx(hDC, rect.right, -rect.bottom, NULL);  
//  
SelectObject(hDC, GetStockObject(BLACK_PEN));  
SelectObject(hDC, GetStockObject(NULL_BRUSH));  
//  
for (int i = 50; i < 500; i += 50)  
    Ellipse(hDC, -i, -i, i, i);
```

Пример центрирования изображения (окончание)

Пример (окончание)

```
//  
EndPaint(hWnd, &paint_s);  
break;  
}  
case WM_DESTROY:  
    PostQuitMessage(0);        // отправить WM_QUIT в очередь  
    break;  
default:    // все остальные сообщения обрабатываются системой  
    return DefWindowProc(hWnd, uMessage, wParam, lParam);  
}    // switch (uMessage)  
//  
return 0;  
}    // WindowProcedure()
```

Функции MoveToEx() и LineTo()

Определение MoveToEx()

```
BOOL MoveToEx(  
    __in    HDC    hDC,  
    __in    int    nX,  
    __in    int    nY,  
    __out   LPPOINT lpPoint  
);
```

Определение LineTo()

```
BOOL LineTo(  
    __in    HDC    hDC,  
    __in    int    nXEnd,  
    __in    int    nYEnd  
);
```

Функция Polygon()

Определение Polygon()

```
BOOL Polygon(  
    __in     HDC          hDC,  
    __in     const POINT *lpcPoints,  
    __in     int         nCount  
);
```

Определение POINT

```
typedef struct tagPOINT {  
    LONG x;  
    LONG y;  
} POINT, *PPOINT;
```

Пример

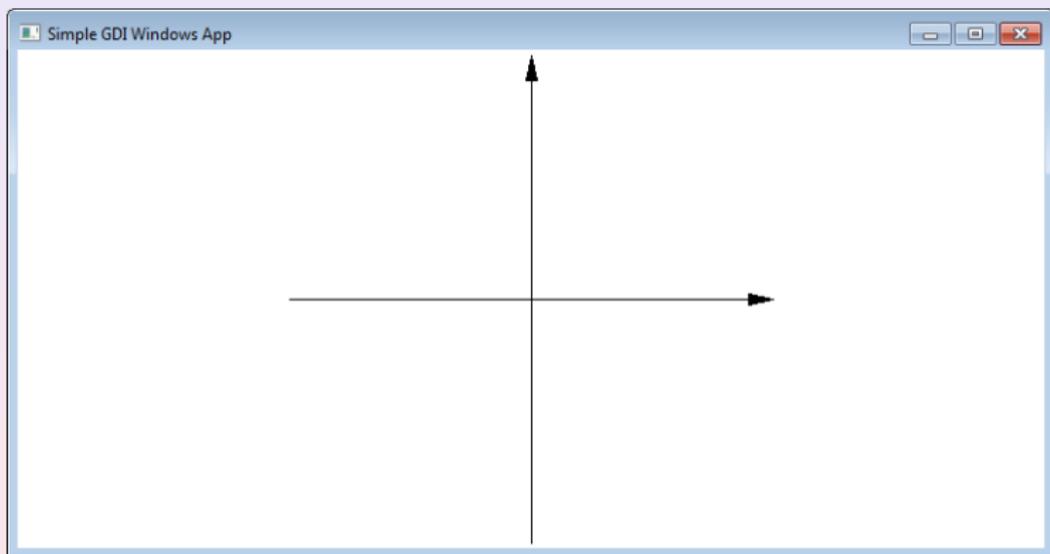


Рис. 4: изображение осей координат

Пример изображения осей

Пример

```
case WM_PAINT:  
{  
    // (Получение контекста отображения и установка способа  
    // преобразования координат)  
    // ...  
    //  
    SelectObject(hDC, GetStockObject(BLACK_PEN));  
    SelectObject(hDC, GetStockObject(BLACK_BRUSH));  
    //  
    MoveToEx(hDC, -490, 0, NULL);  
    LineTo(hDC, 490, 0);  
    MoveToEx(hDC, 0, -490, NULL);  
    LineTo(hDC, 0, 490);  
}
```

Пример изображения осей (окончание)

Пример (окончание)

```
const POINT caArrowH[] =
    { { 490, 0 }, { 440, -10 }, { 440, 10 } };
Polygon(hDC, caArrowH, 3);
//
const POINT caArrowV[] =
    { { 0, 490 }, { -10, 440 }, { 10, 440 } };
Polygon(hDC, caArrowV, 3);
//
EndPoint(hWnd, &paint_s);
break;
} // case WM_PAINT
```

Функция DrawText()

Определение DrawText()

```
int DrawText(  
    __in    HDC    hDC,  
    __inout LPCTSTR lpchText,  
    __in    int   nCount,  
    __inout LPRECT lpRect,  
    __in    UINT   uFormat  
);
```

DT_LEFT	DT_TOP	DT_NOPREFIX
DT_CENTER	DT_VCENTER	DT_SINGLELINE
DT_RIGHT	DT_BOTTOM	DT_WORDBREAK
		DT_NOCLIP
		DT_CALCRECT

Таблица 6: основные форматы текста

Пример

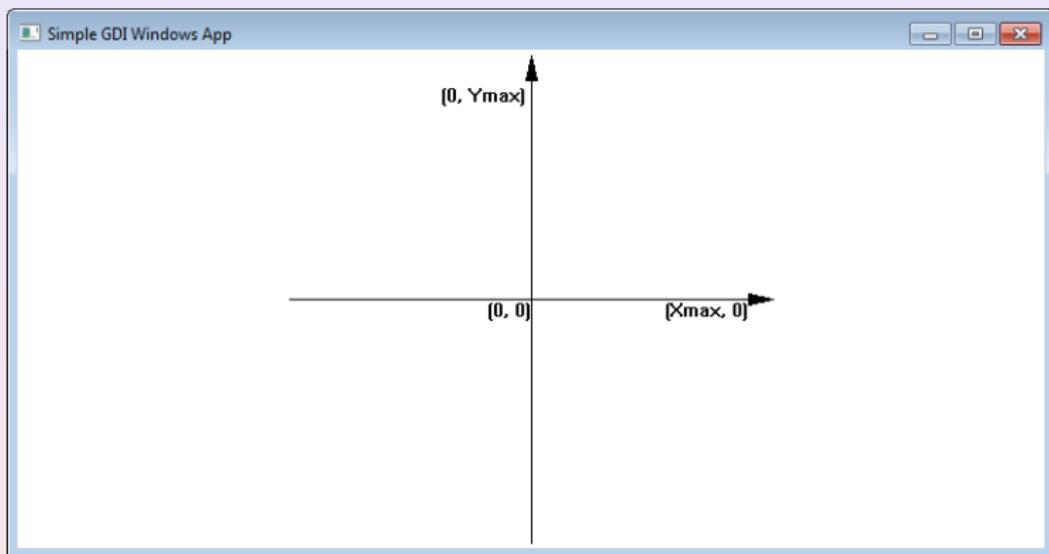


Рис. 5: надписи на осях координат

Пример вывода надписей

Пример

```
case WM_PAINT:  
{  
    // (Как и раньше) ...  
    //  
    TCHAR tszStart[] = TEXT("(0, 0)");  
    TCHAR tszStopX[] = TEXT("(Xmax, 0)");  
    TCHAR tszStopY[] = TEXT("(0, Ymax)");  
    //  
    RECT rectS = { -10, 0, 0, -10 };  
    RECT rectX = { 430, 0, 440, -10 };  
    RECT rectY = { 0, 430, -10, 440 };  
    //  
}
```

Пример вывода надписей (окончание)

Пример (окончание)

```
SetBkMode(hDC, TRANSPARENT);  
DrawText(  
    hDC, tszStart, -1, &rectS, DT_NOCLIP | DT_RIGHT | DT_TOP);  
DrawText(  
    hDC, tszStopX, -1, &rectX, DT_NOCLIP | DT_RIGHT | DT_TOP);  
DrawText(  
    hDC, tszStopY, -1, &rectY, DT_NOCLIP | DT_RIGHT | DT_TOP);  
//  
EndPaint(hWnd, &paint_s);  
break;  
}    // case WM_PAINT
```

Функция CreateFontIndirect()

Определение

```
HFONT CreateFontIndirect(  
    __in    const LOGFONT *lpLogFont  
);
```

Определение LOGFONT

```
typedef struct tagLOGFONT {  
    LONG   lfHeight;  
    // ...  
    BYTE   lfCharSet;  
    // ...  
    TCHAR  lfFaceName[LF_FACESIZE];  
} LOGFONT, *PLOGFONT;
```

Функция GetTextExtentPoint32()

Определение

```
BOOL GetTextExtentPoint32(  
    __in    HDC    hDC,  
    __in    LPCTSTR lpctszString,  
    __in    int    nSize,  
    __out   LPSIZE lpSize  
);
```

Определение SIZE

```
typedef struct tagSIZE {  
    LONG cx;  
    LONG cy;  
} SIZE, *PSIZE;
```

Пример

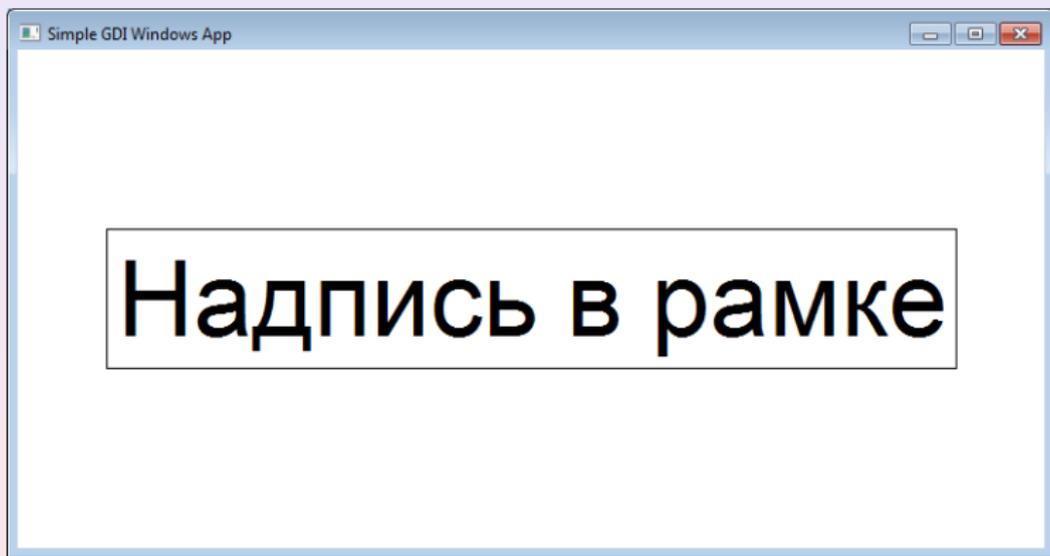


Рис. 6: вывод надписи в рамке

Пример вывода надписи в рамке

Пример

```
case WM_PAINT:  
{  
    // (Получение контекста и установка координат) ...  
    SelectObject(hDC, GetStockObject(BLACK_PEN));  
    SelectObject(hDC, GetStockObject(NULL_BRUSH));  
    //  
    LOGFONT font = { 240, 0 };  
    font.lfCharSet = RUSSIAN_CHARSET;  
    lstrcpy(font.lfFaceName, TEXT("Arial"));  
    HFONT hFont = CreateFontIndirect(&font);  
    HGDIOBJ hFontOld = SelectObject(hDC, (HGDIOBJ) hFont);  
    //
```

Пример вывода надписи в рамке (продолжение)

Пример (продолжение)

```
TCHAR tszText[] = TEXT("Надпись в рамке");  
SIZE sizeB;  
GetTextExtentPoint32(hDC, tszText, 15, &sizeB);  
RECT rectB =  
{  
    -sizeB.cx / 2,  sizeB.cy / 2,  
    sizeB.cx / 2,  -sizeB.cy / 2  
};
```

Пример вывода надписи в рамке (продолжение)

Пример (продолжение)

```
DrawText(  
    hDC, tszText, -1, &rectB,  
    DT_CENTER | DT_VCENTER | DT_NOCLIP | DT_SINGLELINE |  
    DT_CALCRECT);  
DrawText(  
    hDC, tszText, -1, &rectB,  
    DT_CENTER | DT_VCENTER | DT_NOCLIP | DT_SINGLELINE );  
Rectangle(  
    hDC, rectB.left - 20, rectB.top + 20,  
    rectB.right + 20, rectB.bottom - 20);
```

Пример вывода надписи в рамке (окончание)

Пример (окончание)

```
//  
SelectObject(hDC, hFontOld);  
DeleteObject(hFont);  
//  
EndPaint(hWnd, &paint_s);  
break;  
}
```

Функции CreateCompatibleDC() и BitBlt()

Определение CreateCompatibleDC()

```
HDC CreateCompatibleDC(  
    __in    HDC    hdc  
);
```

CreateCompatibleBitmap()

```
HBITMAP CreateCompatibleBitmap(  
    __in    HDC    hdc,  
    __in    int    nWidth,  
    __in    int    nHeight  
);
```

Определение BitBlt()

```
BOOL BitBlt(  
    __in    HDC    hdcDest,  
    __in    int    nXDest,  
    __in    int    nYDest,  
    __in    int    nWidth,  
    __in    int    nHeight,  
    __in    HDC    hdcSrc,  
    __in    int    nXSrc,  
    __in    int    nYSrc,  
    __in    DWORD   dwRop  
);
```

Коды растровых операций

Код	Действие
SRCCOPY	n_{src}
SRCAND	$n_{src} \wedge n_{dst}$
SRCPAINT	$n_{src} \vee n_{dst}$
SRCINVERT	$n_{src} \otimes n_{dst}$
SRCERASE	$n_{src} \wedge \neg n_{dst}$
NOTSRCCOPY	$\neg n_{src}$
NOTSRCERASE	$\neg (n_{src} \vee n_{dst})$

Таблица 7: основные растровые операции при копировании изображения

Пример

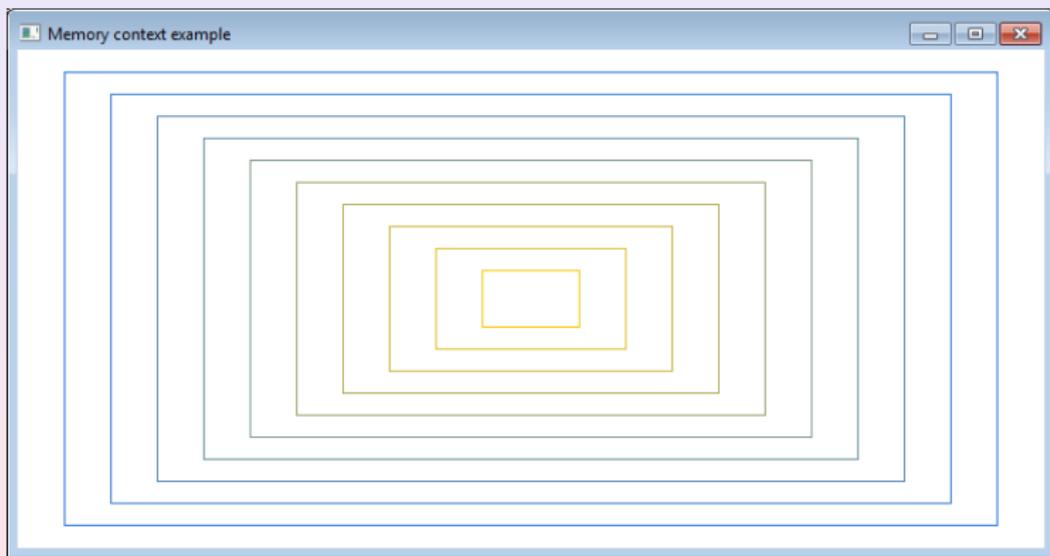


Рис. 7: цветные прямоугольники

Пример использования контекста памяти

Пример

```
int APIENTRY WinMain(
    HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPTSTR lpszCmdLine, int nCmdShow)
{
    WNDCLASSEX wnd_class;
    wnd_class.cbSize = sizeof (WNDCLASSEX);
    wnd_class.style = CS_HREDRAW | CS_VREDRAW;
    // ...
    wnd_class.hbrBackground = (HBRUSH) NULL;
    // ...
    return messages.lParam;
}
```

Пример использования контекста памяти (продолжение)

Пример (продолжение)

```
LRESULT CALLBACK WindowProcedure(  
    HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)  
{  
    static HDC s_hWindowDC = NULL;  
    static HDC s_hMemDC = NULL;  
    switch (uMsg)  
    {  
        case WM_CREATE:  
        {  
            s_hWindowDC = GetDC(hWnd);  
            s_hMemDC = CreateCompatibleDC(s_hWindowDC);  
            break;  
        }  
    }  
}
```

Пример использования контекста памяти (продолжение)

Пример (продолжение)

```
case WM_SIZE:
{
    WORD wWidth = LOWORD(lParam);
    WORD wHeight = HIWORD(lParam);
    HBITMAP hBitmap = CreateCompatibleBitmap(
        s_hWindowDC, wWidth, wHeight);
    HGDIOBJ hObject = SelectObject(s_hMemDC, hBitmap);
    DeleteObject(hObject);
    //
    RECT rect = { 0, 0, wWidth, wHeight };
    FillRect(s_hMemDC, &rect, (HBRUSH) RGB(255, 255, 255));
    //
}
```

Пример использования контекста памяти (продолжение)

Пример (продолжение)

```
const int cnDWidth = wWidth / 22;  
const int cnDHeight = wHeight / 22;  
int nLeft = 0;  
int nTop = 0;  
int nRight = wWidth;  
int nBottom = wHeight;  
int nR = 0;  
int nG = 100;  
int nB = 255;
```

Пример (продолжение)

```
for (int i = 0; i < 10; ++ i)  
{  
    nLeft += cnDWidth;  
    nTop += cnDHeight;  
    nRight -= cnDWidth;  
    nBottom -= cnDHeight;  
    nR += 25;  
    nG += 10;  
    nB -= 25;
```

Пример использования контекста памяти (продолжение)

Пример (продолжение)

```
//  
HPEN hPen = CreatePen(PS_SOLID, 0, RGB(nR, nG, nB));  
HGDIOBJ hOldPen = SelectObject(s_hMemDC, hPen);  
DeleteObject(hOldPen);  
//  
Rectangle(  
    s_hMemDC, nLeft, nTop, nRight, nBottom);  
//  
}    // for (int i = 0; i < 10; ++ i)  
//  
break;  
//  
}    // case WM_SIZE
```

Пример использования контекста памяти (продолжение)

Пример (продолжение)

```
case WM_PAINT:
{
    PAINTSTRUCT paint_s;
    HDC hDC = BeginPaint(
        hWnd, &paint_s);
    //
    RECT rect;
    GetClientRect(hWnd, &rect);
    //
```

Пример (продолжение)

```
BitBlt(
    hDC,           // hDCDest
    0,             // nXDest
    0,             // nYDest
    rect.right,   // nWidth
    rect.bottom,  // nHeight
    s_hMemDC,     // hDCSrc
    0,             // nXSrc
    0,             // nYSrc
    SRCCOPY);     // dwRop
```

Пример использования контекста памяти (окончание)

Пример (продолжение)

```
//  
EndPaint(hWnd, &paint_s);  
//  
break;  
//  
} // case WM_PAINT  
case WM_DESTROY:  
//  
PostQuitMessage(0);  
//  
break;
```

Пример (окончание)

```
//  
default:  
//  
return DefWindowProc(  
    hWnd, uMsg,  
    wParam, lParam);  
}  
//  
return 0;  
//  
} // WindowProcedure()
```