

Интерактивная Компьютерная Графика

Часть 7-3

(Программирование на WinAPI)

Определение

API (Application Programming Interfaces):

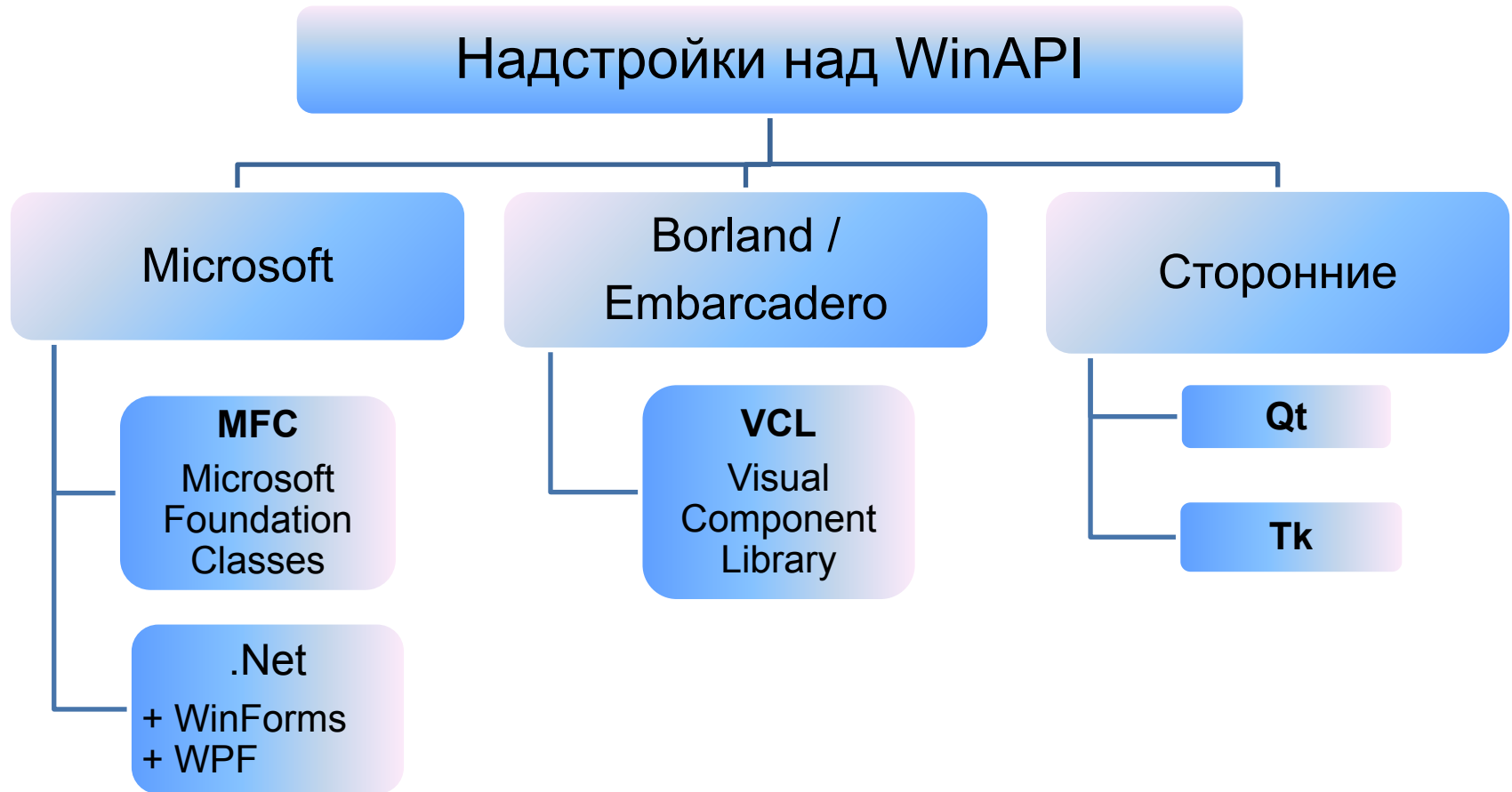
- интерфейс программирования приложений (интерфейс прикладного программирования)
- набор готовых классов, функций, структур и констант, предоставляемых для использования приложению

Windows API – общее наименование целого набора базовых функций интерфейсов для программирования приложений операционных систем семейств Windows

Надстройки над WinAPI :

- Active Template Library (ATL)
- Windows Template Library (WTL)
- Microsoft Foundation Classes (MFC)
- .Net с подсистемами WinForms и WPF

Надстройки



Объекты «кисть», «перо» «декриптор»

Дескриптор – контекст (описатель) устройства (объекта, окна)

```
dc = GetDC ()
```

```
hdc = BeginPaint () // хэндл («указатель») на dc
```

Кисть используется для рисования объектов с заливкой

```
HBRUSH hBrush // создание объекта-кисть
```

```
CreateSolidBrush ( RGB(255,0,0) ) // сплошная заливка красным цветом
```

```
CreateHatchBrush ( HS_CROSS, RGB(255,0,0)) // закрашивание в клеточку
```

```
SelectObject ( hdc, hBrush) // выбор активной кисти
```

Перо используется для рисования линий

```
HPEN hPen // создание объекта-перо
```

```
CreatePen ( PS_SOLID, width, RGB(r,g,b)) // непрерывная красная линия
```

```
CreatePen ( PS_DASH , width, RGB(r,g,b)) // пунктирная красная линия
```

```
SelectObject ( hdc, hPen ) // выбор активного пера
```

Функции рисования

SetPixel (hdc, x,y, color) // Закрашивание пикселя (x,y) цветом color

MoveToEx (hdc, x, y, point) // Изменение текущей позиции пера

LineTo (hdc, x, y) // Рисование линии от текущей позиции до (x,y)

Rectangle (hdc, x0,y0, x1,y1) // Рисование прямоугольника

Ellipse (hdc, x0,y0, x1,y1) // Рисование эллипса

Polyline (*) // Рисование ломаной по точкам points

Arc (*) // Рисование дуги эллипса или окружности

RoundRect (*) // Рисование скругленного прямоугольника

Chord (*) // Рисование сегмента эллипса

Pie (*) // Рисование сектора эллипса

Пример рисования (квадрат 100x100 с позиции клика)

```
#include <windows.h>
```

```
#include <math>
```

```
LRESULT CALLBACK WindowProc (HWND, UINT, WPARAM, LPARAM );
```

```
HDC hDc;
```

```
HWND hWnd; // Дескриптор окна
```

```
LPCSTR AppName = «Пример рисования квадрата»;
```

```
void CreateWindowClass ( HINSTANCE );
```

```
void CreateWindowMain ( HINSTANCE );
```

Пример рисования

(квадрат 100x100 с позиции клика)

```
int WINAPI WinMain ( HINSTANCE hInstance, HINSTANCE hPrevInstance,
                    LPSTR lpCmdLine, int nCmdShow)
{
    MSG msg; // Объявление структуры типа MSG, для работы с сообщениями

    CreateWindowClass ( hInstance );
    CreateWindowMain ( hInstance );

    hDc = GetDC (hWnd);

    ShowWindow (hWnd, nCmdShow);
    UpdateWindow(hWnd);

    while ( GetMessage ( &msg, NULL, 0, 0) )
    {
        TranslateMessage ( &msg ); // расшифровать сообщение
        DispatchMessage ( &msg ); // передать сообщение дальше
    }

    return msg.wParam;
}
```

Пример рисования

(квадрат 100x100 с позиции клика)

```
void CreateWindowClass ( HINSTANCE hInst)
{
    WNDCLASS wc; // класс окна

    ZeroMemory(&wc, sizeof(wc));

    wc.style = CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc = (WNDPROC)WndProc;
    wc.hInstance = hInst;
    wc.hIcon = LoadIcon ( hInst, IDI_APPLICATION );
    wc.hCursor = LoadCursor ( NULL, IDC_ARROW );
    wc.lpszClassName = AppName;

    int err = RegisterClass(&wc); // Регистрация оконного класса
    if( !err ) exit(0);
}
```


Пример рисования

(квадрат 100x100 с позиции клика)

```
void CreateWindowMain ( HINSTANCE hInst )
{
    hWnd = CreateWindow
    (
        AppName, // Имя класса окна
        AppName, // Заголовок окна
        WS_OVERLAPPEDWINDOW, // Стилль окна
        CW_USEDEFAULT, 0, // Горизонтальная и вертикальная позиции окна
        800, 600, // Ширина и высота окна
        NULL, // Хендл родительского окна
        NULL, // Хендл меню
        hInst, // Идентификатор приложения
        NULL // Дополнительные данные окна
    );
}
```

Пример рисования

(квадрат 100x100 с позиции клика)

```
LRESULT CALLBACK WindowProc (HWND hWnd, UINT message,
                             WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_LBUTTONDOWN:
            int x = LOWORD ( lParam ); // получение x-координаты
            int y = HIWORD ( lParam ); // получение y-координаты

        case WM_PAINT:
            Rectangle (hDc, x, y, x+100, y+100); // прямоугольник с кистью по умолчанию
            break;

        case WM_KEYDOWN:
            if (wParam != VK_ESCAPE)
                break;

        case WM_DESTROY:
            PostQuitMessage (0); // сообщение о нормальном завершении программы
    }

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