

Введение в Windows Forms



C

(язык C в 1972 г. Дэннисом Ритчи)

WinAPI 32

(Windows Application Programming
Interface)

C++/MFC

(язык C++ в 1979 г, Бьярни
Страуструпом,
стандартизирован в 1997)

Visual Basic

Windows Forms

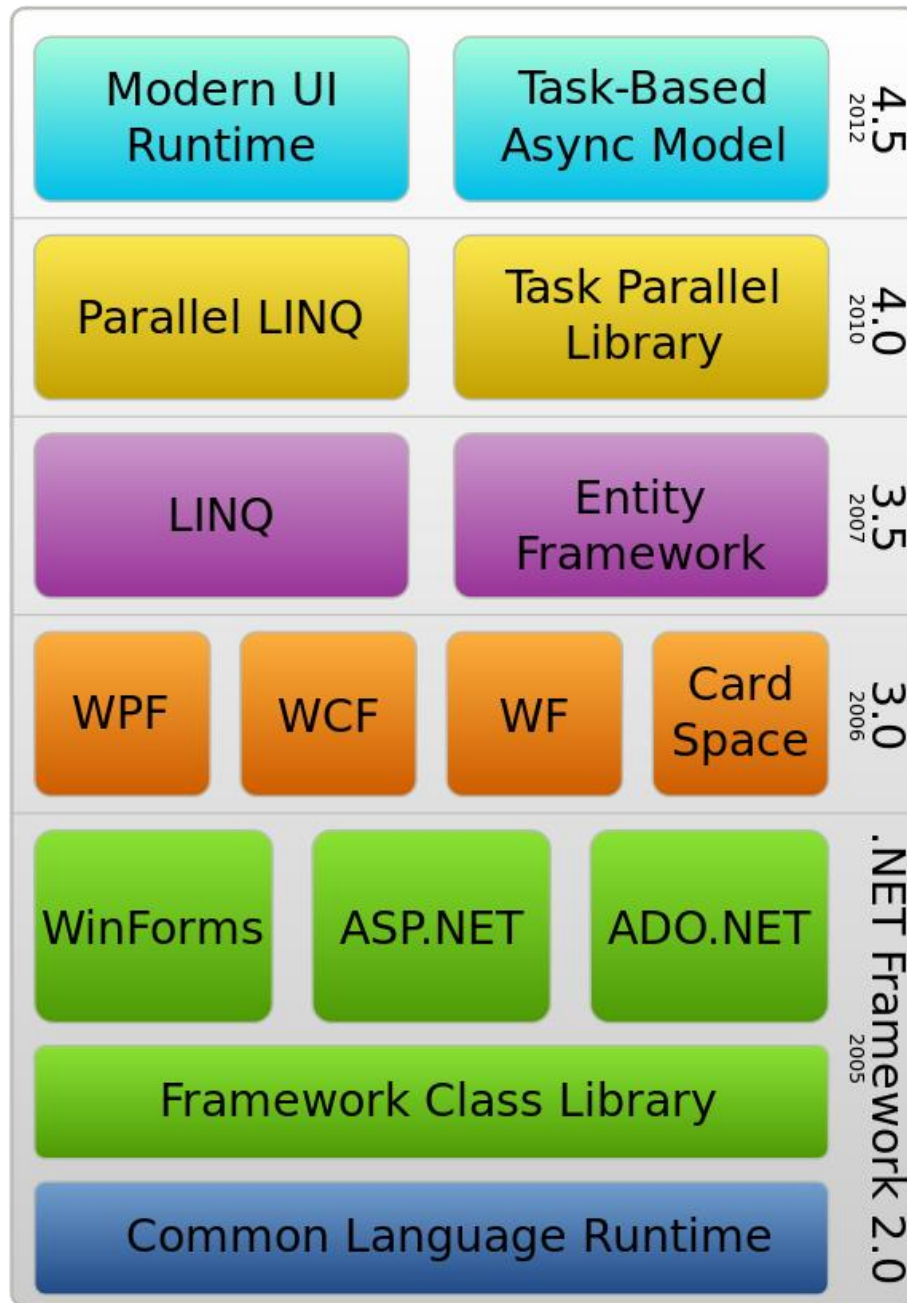
.NET Framework 2.0

11 июля 2005 года

Windows Presentation Foundation

.NET Framework 3.0

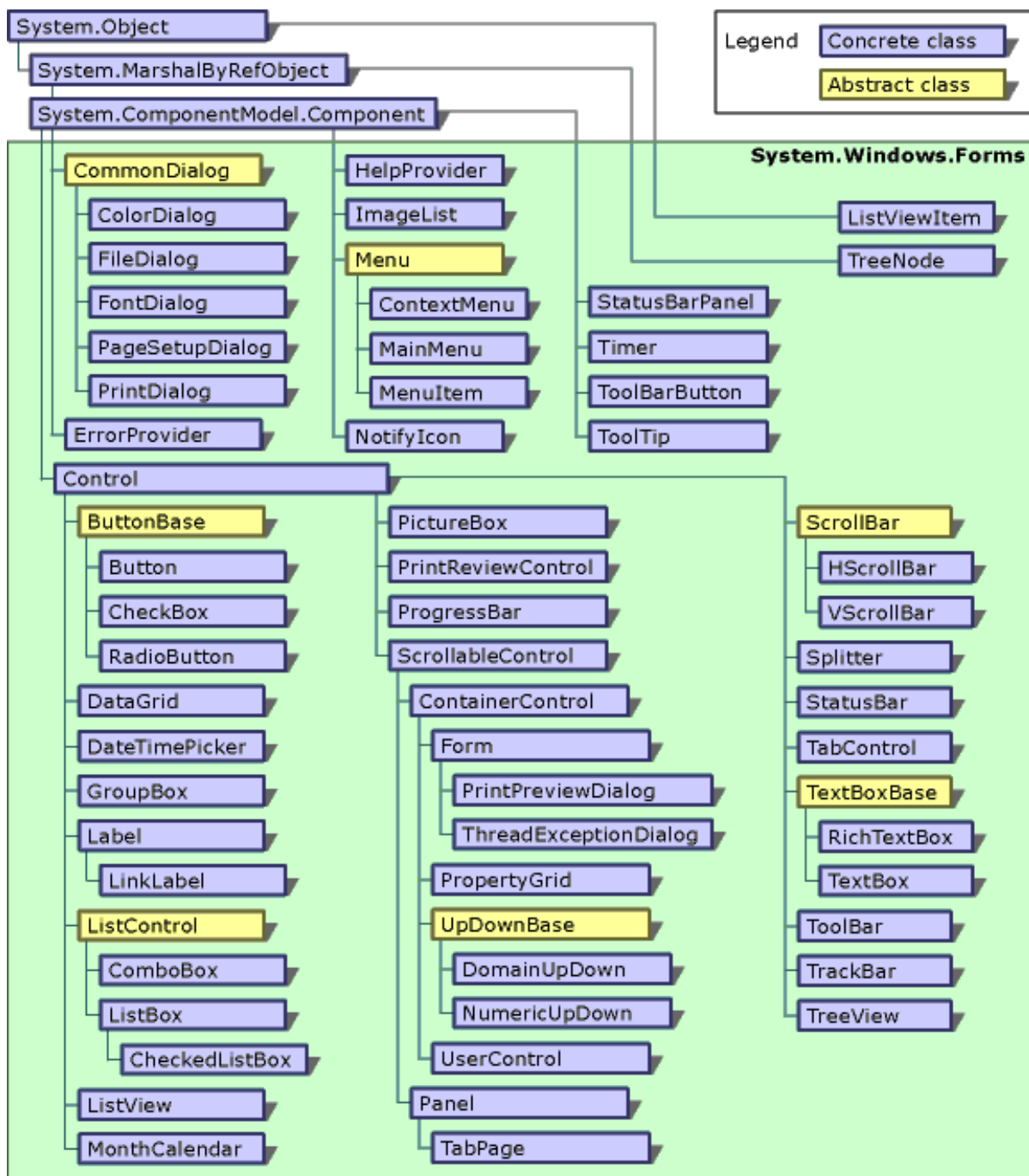
6 ноября 2006 года



Windows Forms — набор классов, отвечающий за создание графического интерфейса пользователя и являющийся частью Microsoft .NET Framework.



Данный набор классов упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде.



```

#include <windows.h>
LRESULT CALLBACK WndProc(HWND hwnd,
                          UINT msg,
                          WPARAM wparam,
                          LPARAM lparam);
INT WINAPI WinMain(HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPSTR cmdline, int cmdshow)
{
MSG msg;
HWND hwnd;
WNDCLASSEX wndclass = { 0 };
wndclass.cbSize = sizeof(WNDCLASSEX);
wndclass.style = CS_HREDRAW | CS_VREDRAW;
wndclass.lpfnWndProc = WndProc;
wndclass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
wndclass.hCursor = LoadCursor(NULL, IDC_ARROW);
wndclass.hbrBackground =
(HBRUSH)GetStockObject(WHITE_BRUSH);
wndclass.lpszClassName = TEXT(«Window1»);
wndclass.hInstance = hInstance;
wndclass.hIconSm = LoadIcon(NULL,
                             IDI_APPLICATION);

RegisterClassEx(&wndclass);
hwnd = CreateWindow(TEXT(«Window1»),
                   TEXT(«Hello World»),
                   WS_OVERLAPPEDWINDOW,
                   CW_USEDEFAULT,
                   0,

```

```

                   CW_USEDEFAULT,
                   0,
                   NULL,
                   NULL,
                   hInstance,
                   NULL);

if( !hwnd )
return 0;
ShowWindow(hwnd, SW_SHOWNORMAL);
UpdateWindow(hwnd);
while( GetMessage(&msg, NULL, 0, 0) )
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
return msg.wParam;
}
LRESULT CALLBACK
WndProc(HWND hwnd, UINT msg,
        WPARAM wparam, LPARAM lparam) {
switch(msg) {
case WM_DESTROY:
PostQuitMessage(WM_QUIT);
break;
default:
return
DefWindowProc(hwnd, msg, wparam, lparam);
}
return 0;}

```

Программа «Hello, World» на языке Си с использованием WinAPI

```

import java.awt.event.*;
import java.awt.*;
class simpleFrame extends Frame
{
public static void main(String[] args)
    {
        simpleFrame a= new
            simpleFrame(" Hello, World ");
    }
simpleFrame(String title)
{
    setTitle(title);
    show();
} }

```

**Программа «Hello, World» на
языке Java**

```

using System.Windows.Forms;
using System;
class Program
{
    [STAThread]
    static void Main()
    {
        Form myForm = new Form();
        myForm.Text = «Hello World»;
        Application.Run(myForm );
    }
}

```

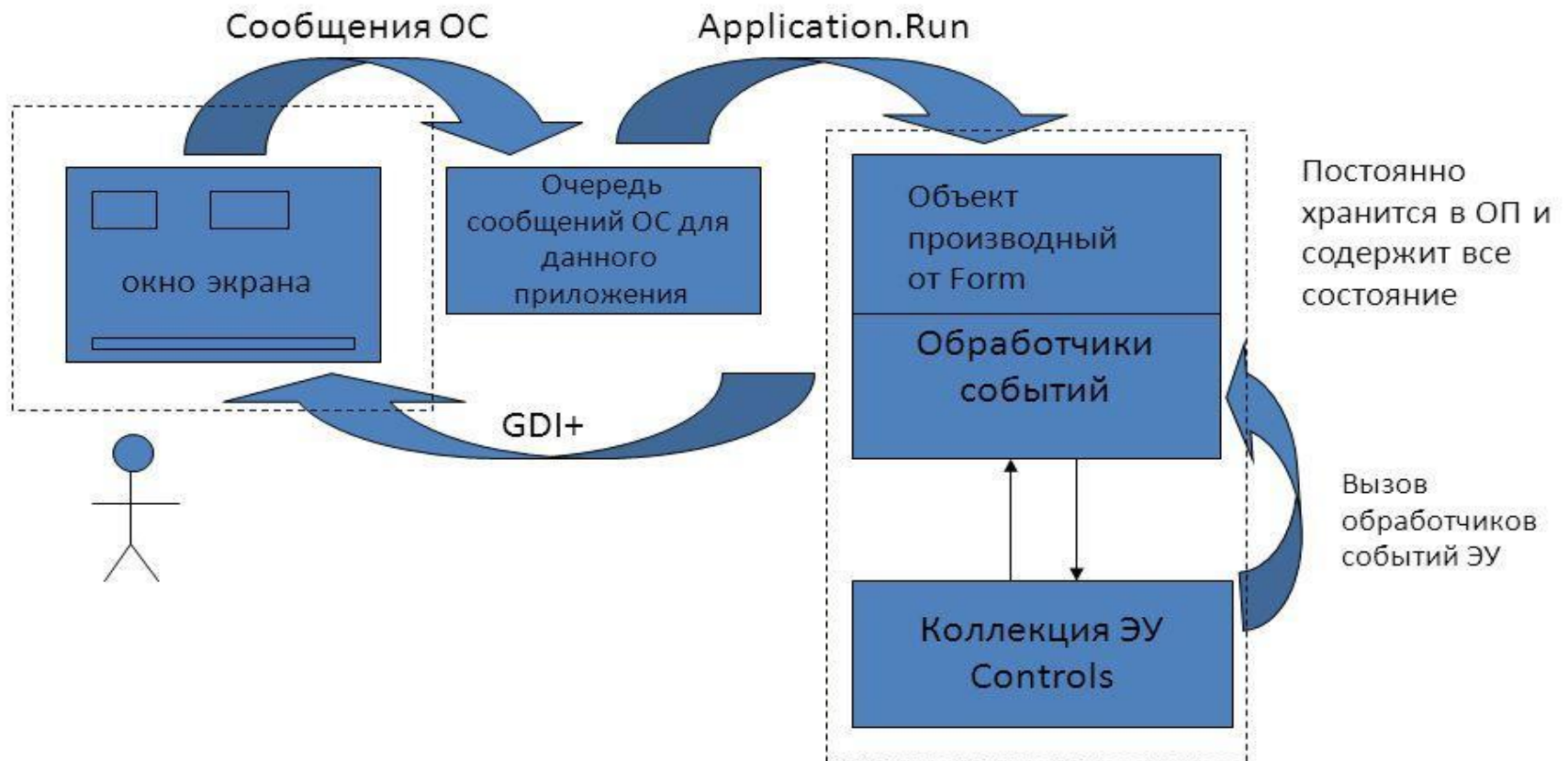
**Программа «Hello, World» на
языке C# платформа .NET**

Windows Forms рассматривается как замена более старой и сложной библиотеке MFC, изначально написанной на языке C++.

С выходом .NET Framework 3.0 (в составе Windows Vista) Windows Forms была заменена каркасом Windows Presentation Foundation (и его языком описания - XAML).

Приложение Windows Forms представляет собой **событийно-ориентированное** приложение, поддерживаемое Microsoft .NET Framework

Модель Windows приложения



Элементы управления, пользовательские элементы управления и формы

Класс **Control** предоставляет основные функциональные возможности для всех элементов управления, отображаемых в Form. Большинство классов в пространстве имен System.Windows.Forms созданы на основе класса Control.

Класс **Form** представляет окно в приложении. Оно включает диалоговые окна, немодальные окна, а также клиентские и родительские окна интерфейса MDI.

На основе класса **UserControl** можно создавать собственные классы элементов управления.

Меню и панели инструментов

Windows Forms включает широкий набор классов, которые позволяют создавать пользовательские панели инструментов и меню, отличающиеся современным обликом и поведением.

ToolStrip, MenuStrip, ContextMenuStrip и **StatusStrip** позволяют создавать панели инструментов, меню, контекстные меню и строки состояния, соответственно.

Элементы управления

Пространство имен **System.Windows.Forms** предоставляет большое количество классов элементов управления, которые позволяют создавать пользовательские интерфейсы с расширенными возможностями.

Некоторые элементы управления предназначены для ввода данных в приложении, например **TextBox** и **ComboBox**.

Другие элементы управления отображают данные приложений, например **Label** и **ListView**.

Макет

Несколько ключевых классов в Windows Forms предназначены для управления расположением элементов управления на экране, то есть на форме или в элементе управления.

- **FlowLayoutPanel** позволяет разместить все элементы управления один за другим,
- **TableLayoutPanel** позволяет определить строки и ячейки для размещения элементов управления по заданной сетке.
- **SplitContainer** позволяет разделить интерфейс на несколько частей с изменяемыми размерами.

Общие диалоговые окна

Операционная система Windows предоставляет ряд основных диалоговых окон, позволяющих обеспечить единообразие пользовательского интерфейса в приложениях Windows при выполнении таких операций как открытие и сохранение файлов, задание цвета шрифта или текста и печать.

Классы **OpenFileDialog** и **SaveFileDialog** предоставляют возможность отображения диалогового окна, в котором пользователь может выполнить поиск файла, а также ввести имя файла, который необходимо открыть или сохранить.

Класс **FontDialog** отображает диалоговое окно для изменения элементов Font, используемого приложением.

Форма

ФОРМА — это аналог диалога, то есть фактически диалоговое окно, на котором возможно размещать элементы управления. С его помощью происходит взаимодействие с пользователем.

Класс Form в .Net Framework отвечает за операции с формой

`System.Object`

`System.MarshalByRefObject`

`System.ComponentModel.Component`

`System.Windows.Forms.Control`

`System.Windows.Forms.ScrollableControl`

`System.Windows.Forms.ContainerControl`

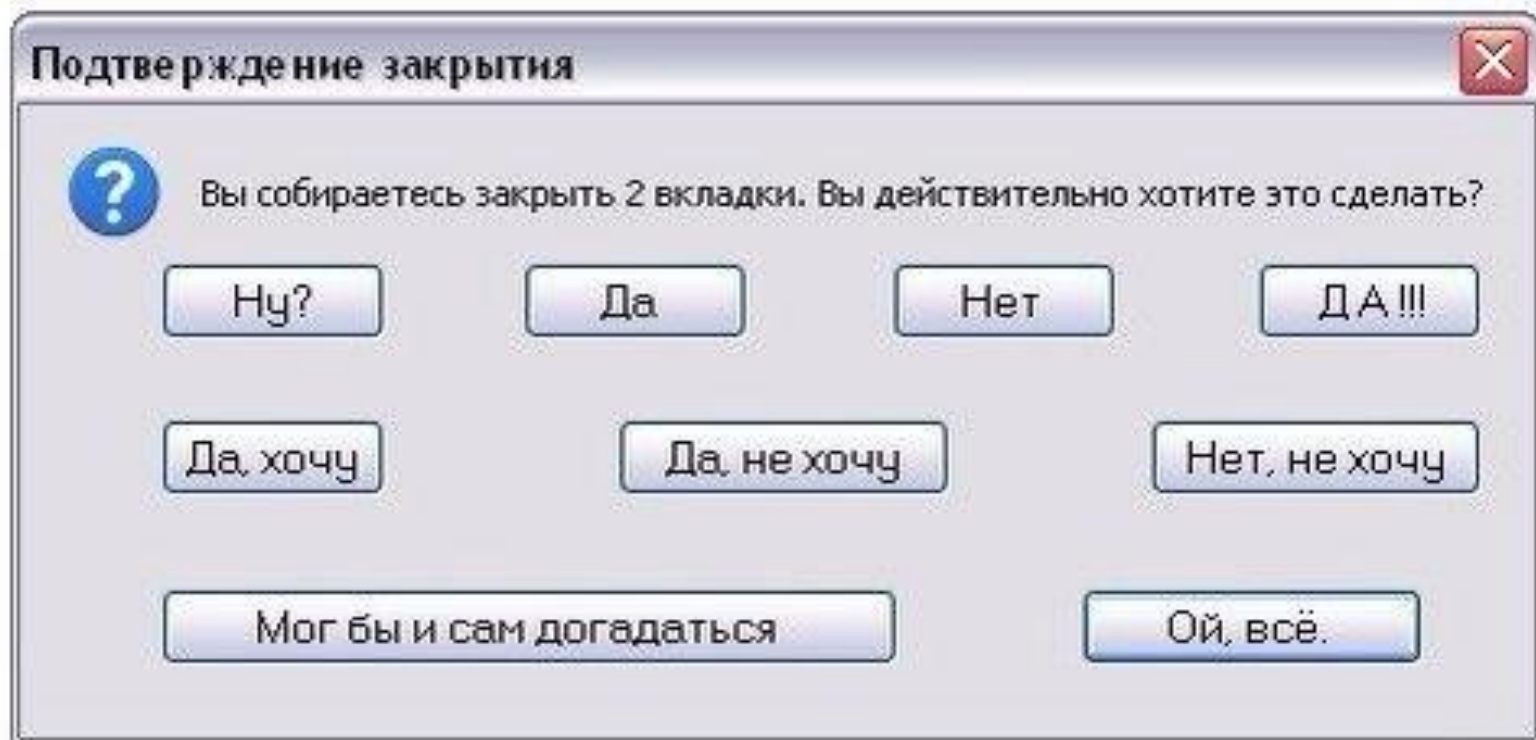
`System.Windows.Forms.Form`

Принципы работы со временем и датой

Для работы со временем и датой в **.Net Framework** используются **структуры** **DateTime** и **TimeSpan**.

- **DateTime** используется для представления **даты и времени**,
- **TimeSpan** для представления **временного промежутка**.

MessageBox



Основные члены класса DateTime

`public static DateTime Now { get; }` получение текущей даты и времени

`public static DateTime Today { get; }` получение текущей даты (время 00:00:00)

Методы для получения строкового представления даты и времени:

`ToLongDateString()`

`ToLongTimeString()`

`ToShortDateString()`

`ToShortTimeString()`

Вариант для получения строкового представления даты и времени в указанном формате:

`DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");`

По отношению к объекту типа **DateTime** можно применять перегруженные операторы, такие как `+, -, ==, <, >, <=, >=, !=`

Основные члены класса **TimeSpan**

Свойства **Days**, **Hours**, **Minutes**, **Seconds**, **Milliseconds** используются для получения соответственно дневной, часовой, минутной, секундной и миллисекундой составляющей **TimeSpan**.