

Высокоуровневые методы информатики и программирования

Лекция 20

Пространство имен
`System.Windows.Forms`

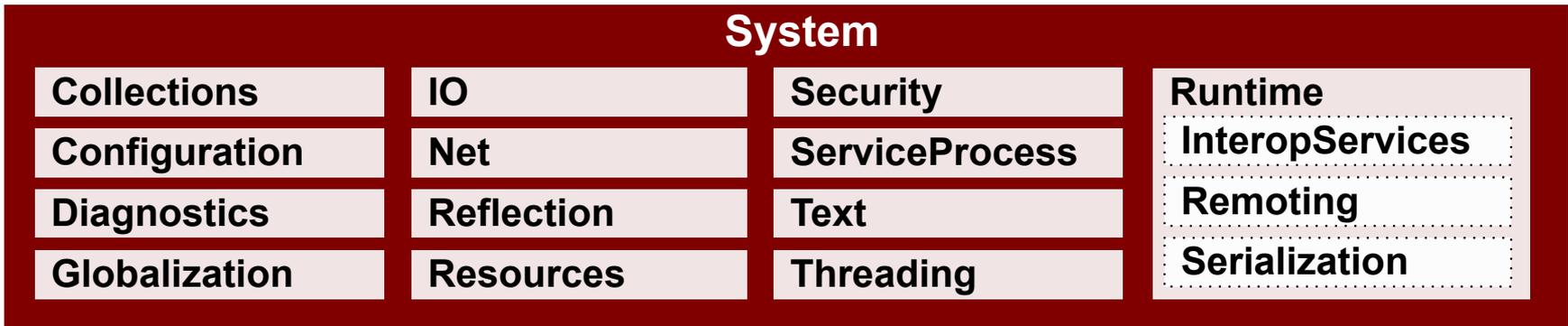
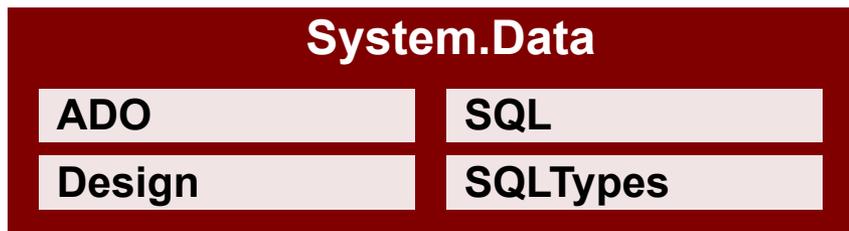
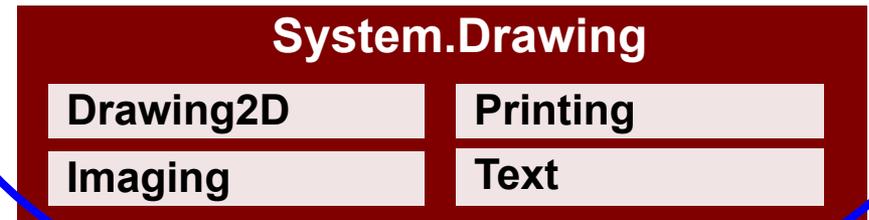
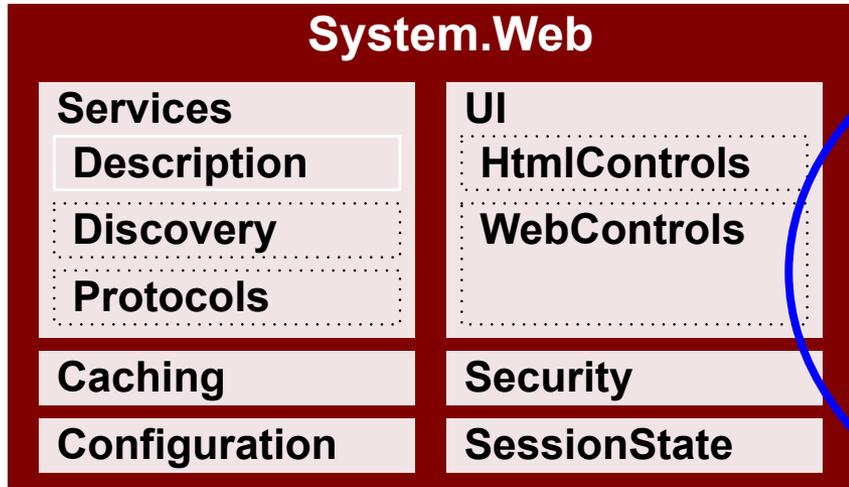
Технологии разработки GUI

- Набор классов пространства `System.Windows.Forms`
- Windows Presentation Foundation (WPF)

Работа с окнами и сообщениями с использованием `System.Windows.Forms`

- На платформе .Net для работы с этой библиотекой Win32 API программист должен использовать классы пространства имен `System.Windows.Forms`
- Для рисования в окне используются классы пространства имен `System.Drawing`

Библиотека FCL (Framework Class Library)



Основные классы пространства имен `System.Windows.Forms`

Класс	Пояснение
<code>Control</code>	базовый класс для окон и элементов упр.
<code>Form</code>	основное окно Windows (контейнер для элементов управления)
<code>Application</code>	статический класс приложения
<code>Menu</code>	класс меню
<code>MainStrip</code>	основное меню
<code>ContextMenuStrip</code>	контекстное меню
<code>MessageBox</code>	статический класс окон сообщений
<code>Timer</code>	работа с таймером
<code>Clipboard</code>	работа с буфером обмена

Класс

System.Windows.Forms.Application

- Статический класс для управления всем приложением в целом
 - Запуск цикла обработки сообщений ОС
 - Завершение работы
 - Получение информации о приложении
 - События
 - о простое приложения
 - о завершении работы
- Методы:
 - Application.Run(ссылка на форму)
 - Exit()
- События:
 - ApplicationExit
 - Idle

Элементы управления - Control

- Специальный класс, включающий окно ОС Windows в качестве поля.
- Элементы данного класса могут:
 - Показывать (рисовать) в заданном месте формы.
 - Обработать сообщения ОС Windows переданные данному окну и формировать события класса о действиях пользователя
 - Работать в двух режимах
 - Design mode – в режиме визуального проектирования интерфейса пользователя
 - Рисовать себя на форме
 - Задавать значения свойств в системе программирования
 - Run-Time mode – при работе программы
 - Менять картинку в зависимости от действий пользователя

Элемент управления

- Класс, объекты которого имеют визуальное представление в виде окна ОС Windows
- Класс Control реализует самую базовую функциональность, требуемую всем классам, которые показывают информацию пользователям. Он обрабатывает действия пользователя с клавиатурой и указывающими устройствами (pointing devices, мышь, джойстик, ...)
- Windows Forms controls
- Использует окружающие свойства (ambient properties). Это такие свойства, которые если не установлены, то они запрашиваются у родительского элемента управления.

Базовый класс Control - элемент управления (окно)

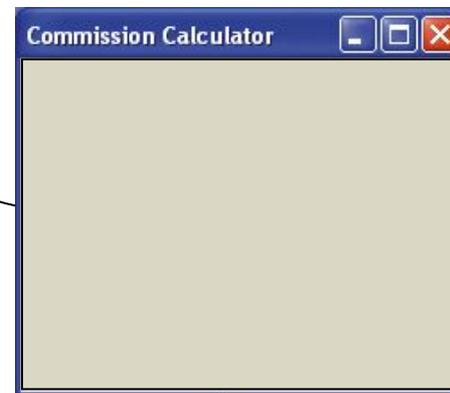
- включает в качестве поля номер окна `hwnd`
- используя `hwnd` и `WinAPI` может управлять своим окном
- получает и обрабатывает события от ОС Windows
- формирует новые события для классов пользователей – клиентов
- поддерживает два режима работы:
 - Design mode
 - Run-time mode

```
Class A : Control
long hwnd = 1024;

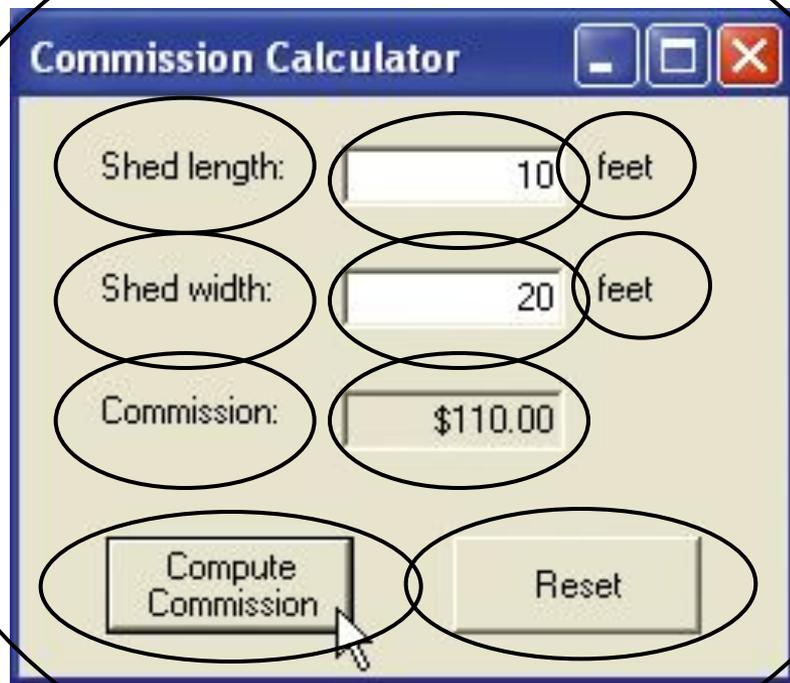
void Show ()
{ ::Show (hwnd, SC_MIN)}

// обработчики событий
```

`hwnd = 1024;`



Основной класс операционной системы window



- Создает и работает с окнами - ОС
- Каждое окно имеет номер – handle of window (hwnd)
- с КАЖДЫМ ОКНОМ связан обработчик событий – оконная процедура (wndProc)

Некоторые базовые свойства класса System.Windows.Forms.Control

- **Handle** – номер окна Windows
- **BackColor, ForeColor** – фоновый цвет и цвет рисования
- **Left, Top, Right, Bottom** – координаты точек диагонали
- **Height, Width** – высота и ширина
- **Size** – размер окна
- **ContextMenu** – ссылка на контекстное меню
- **Cursor** - ссылка на курсор
- **Name** – имя элемента для работы системы разработки
- **Parent** – ссылка на родительское окно
- **Text** – заголовок
- **Font** – используемый шрифт
- **Controls** – список других элементов управления (если контейнер)
- **TabIndex** – порядок перехода при нажатии клавиши Tab
- **Focused** – имеет ли окно фокус ввода (get).
- **Anchor** – связывание со сторонами родительской формы.
- **Dock** – присоединение к сторонам родительской формы.

Некоторые методы базового класса System.Windows.Forms.Control

- `Show()` – показать окно на экране;
- `Hide()` - спрятать окно (убрать с экрана);
- `OnEvent()` – обработать заданное событие;
- `CreateGraphics()` – создать элемент класса `Graphics` для рисования в окне;
- `Update()` – обновить содержание окна;
- `Invalidate()` – объявить содержание окна неправильным, что заставляет ОС отправить событие `WM_PAINT` о необходимости перерисовки содержания окна.

Основные события классов элементов

- Общие события
 - Paint
 - GetFocuse
 - LostFocuse
- Работа с мышкой
 - Click,
 - MouseDown, MouseEnter, MouseHover, MouseLeave, MouseMove MouseUp, MouseWheel
- Работа с клавиатурой
 - KeyDown,
 - KeyUp,
 - KeyPress

Обработка событий

- Для обработки событий нужно иметь
 - Делегата, который описывает метод, обрабатывающий событие.
 - Класс, который содержит событие.

Сигнатура обработчика событий

- Все события классов FCL имеют одинаковую сигнатуру (состав параметров)
- Состав параметров
 - Ссылка на объект, сгенерировавший событие
`Object`
 - Объект содержащий параметры события
`EventArgs` (или производный от него)

```
public EventHandler (Object sender, System.EventArgs e)
```

Делегат `EventHandler`

- Описывает метод, который будет обрабатывать события, не имеющие специальных данных об этом событии

```
public delegate void EventHandler ( Object sender, EventArgs e ) ;
```

- **EventArgs** – базовый класс (пустой) для классов, которые содержат данные, описывающие произошедшее событие.
- Для передачи данных, описывающих событие нужно создать производный класс от класса `System.EventArgs`.

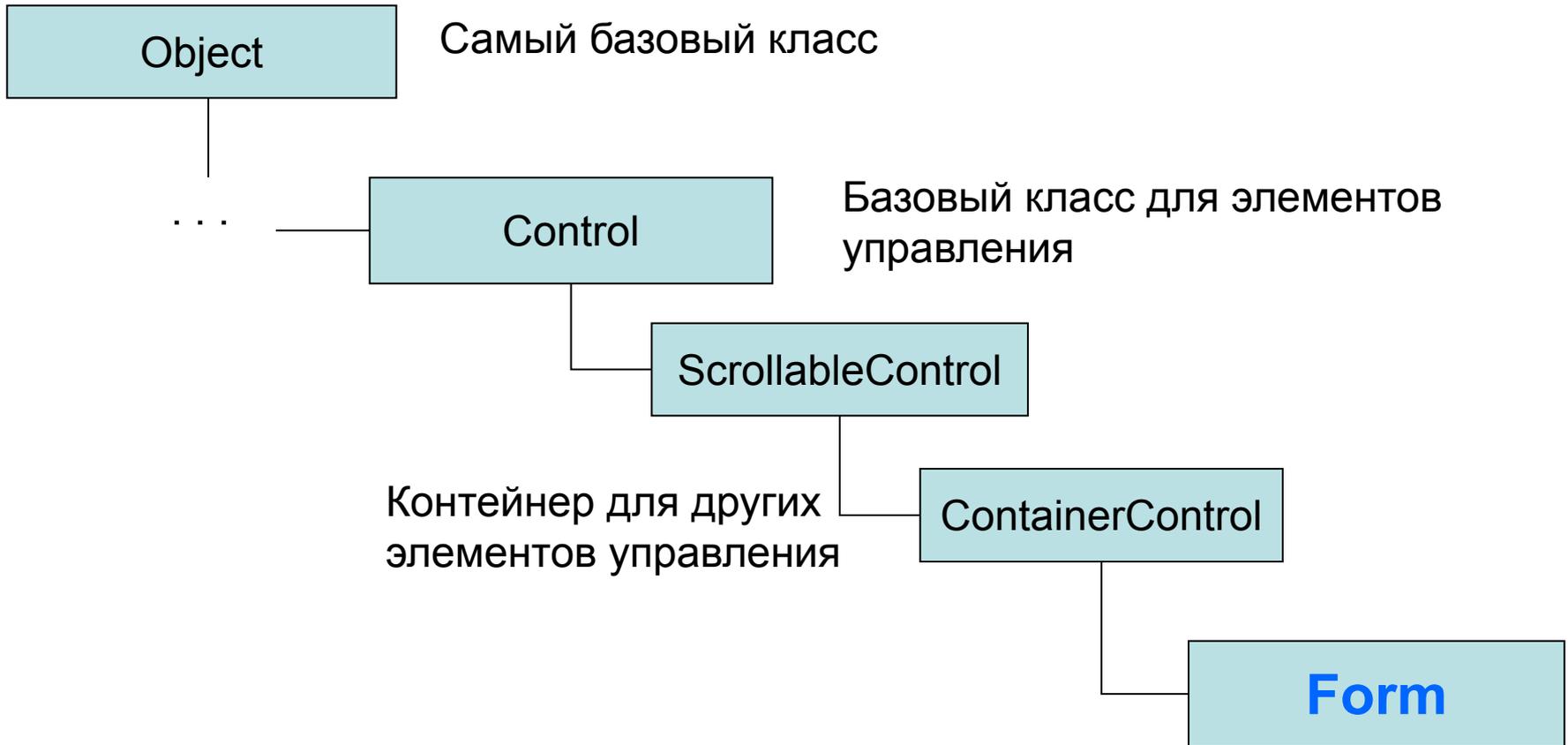
Класс основного окна программы
`System.Windows.Forms.Form`

Класс

System.Windows.Forms.Form

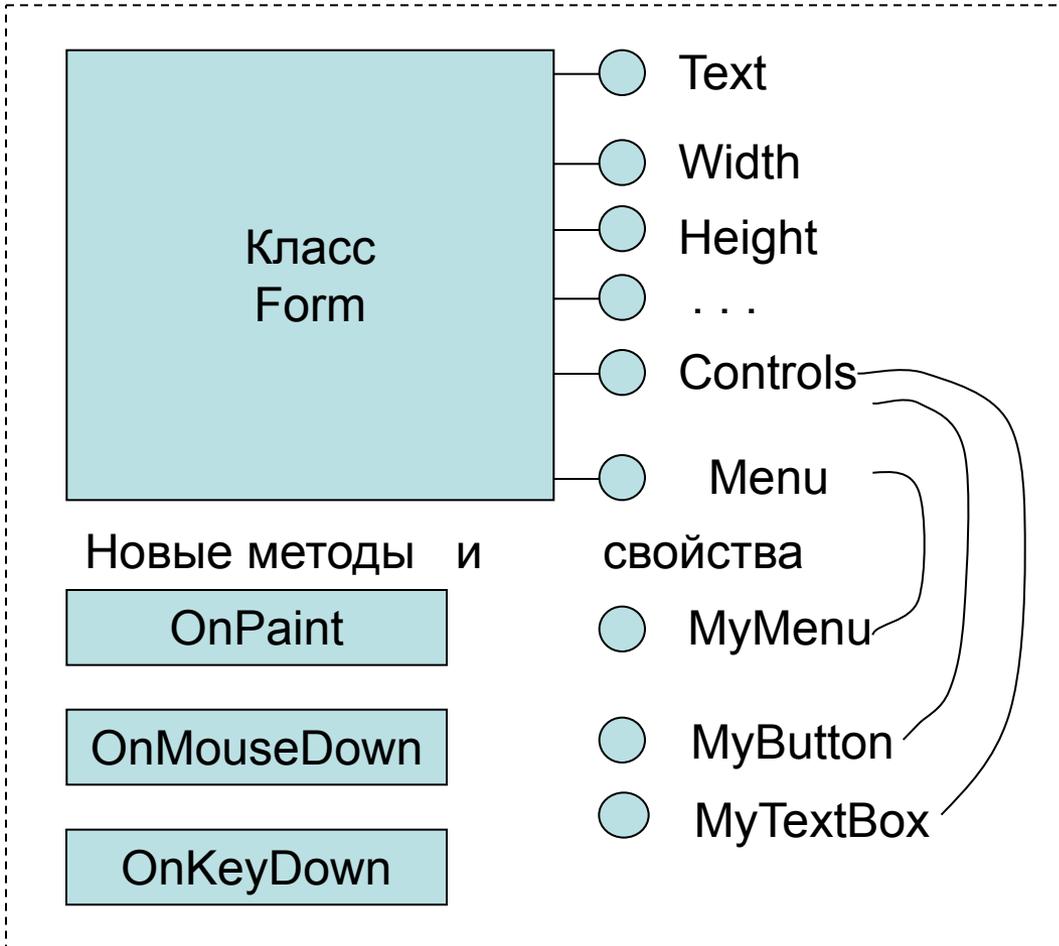
- Форма (Form) это специальный класс, который позволяет пользователю создавать и показывать окно ОС Windows.
- Это окно может быть:
 - Основным окном программы;
 - Диалоговыми окнами программы.
- Для работы с окнами необходимо создать и показать объект класса Form
- Класс Form имеет множество свойств, методов, событий.

Наследственность класса Form



Структура производной формы

Производный от формы класс



События подключаются:

1. event += new (delegate)
2. Используя перегружаемые функции базового класса

Жизненный цикл объектов класса Form

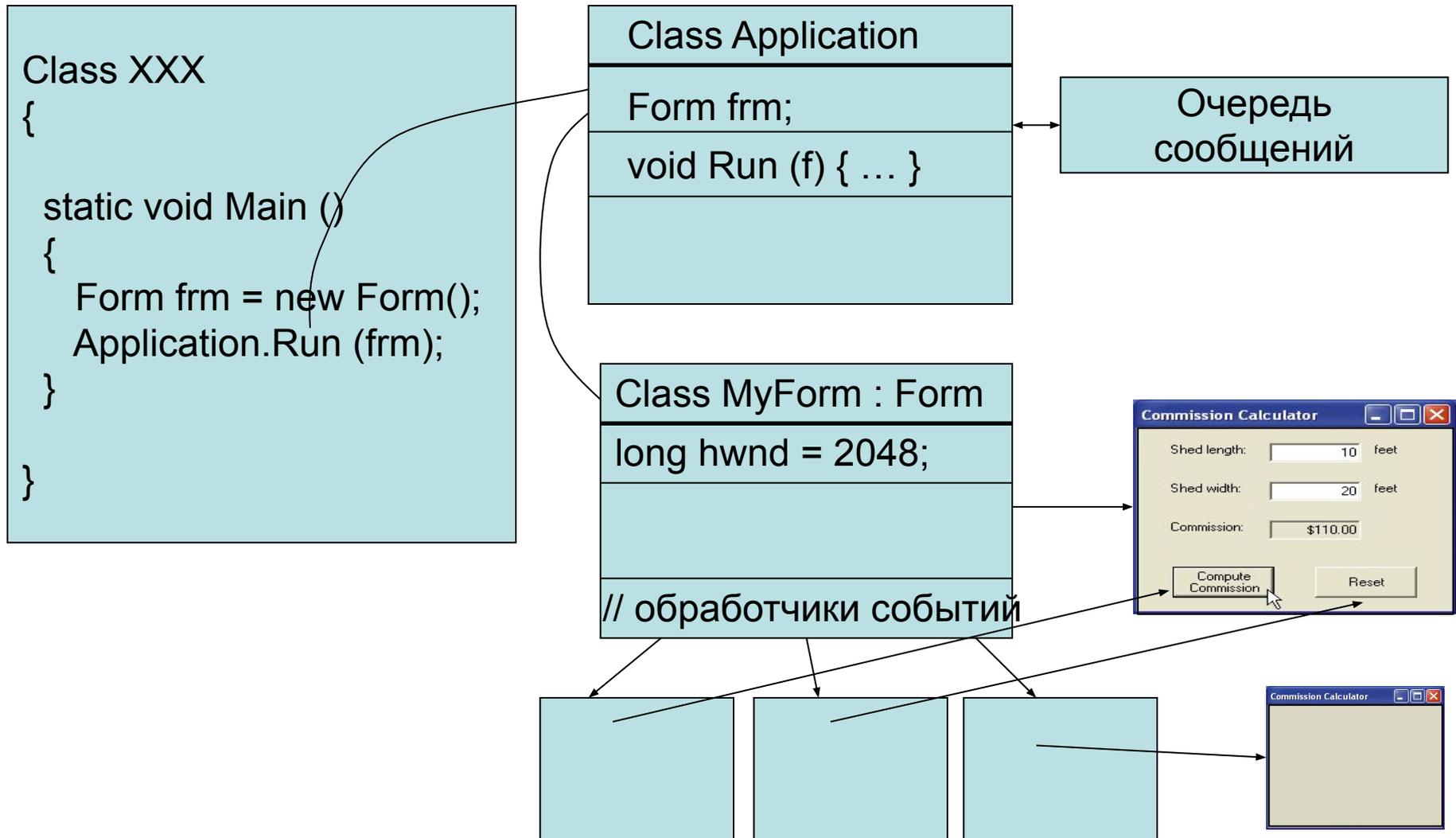
- Жизненный цикл элемента
 - Load
 - Activated
 - Deactivated
 - Close
- Дополнительные события

Класс `System.Windows.Forms.Application`

- Статический класс для управления всем приложением в целом
 - Запуск цикла обработки сообщений ОС
 - Завершение работы
 - Получение информации о приложении
 - События
 - о простое приложения
 - о завершении работы
- Метод передачи сообщений из очереди в объект класса `Form`

`Application.Run` (ссылка на форму);

Структура программы с GUI интерфейсом



Последовательность создания программы с GUI

- Создать класс производный от класса Form.
- Задать в качестве полей класса ссылки на элементы управления, которые будут показаны на форме.
- В конструкторе класса, создать экземпляры элементов управления и установить их свойства.
- Добавить элементы управления к коллекции Controls класса Form.
- Добавить обработчики событий формы или элементов управления, на которые должна реагировать программа с сигнатурой .
void EventHandler (object o, EventArgs ea)
- Связать обработчики с событиями элементов управления или формы.
- Написать класс, который содержит статический метод Main:
`static public void Main()`
- В этом методе создать экземпляр описанного производного от Form класса.
- Запустить цикл обработки событий ОС Windows (указать ссылку на объект производный от Form).

Пример создания программы с GUI

```
class Program
{
    static void Main(string[] args)
    {
        MyForm frm = new MyForm();
        Application.Run(frm);
    }
}
```

```
class MyForm : Form
{
    Button btn;
    public MyForm( )
    {
        this.Text = "Sample Windows Program";
        this.Cursor = Cursors.Cross;

        btn = new Button();
        btn.Text = "Click me!";
        btn.Top = 100; btn.Left = 100;
        this.Paint += new EventHandler (OnPaint);
        this.Controls.Add(btn);
    }

    protected override void Paint(PaintEventArgs pea)
    {
        Graphics grfx = pea.Graphics;
        grfx.DrawString("Привет!", Font, Brushes.Black, 0, 0);
    }
}
```

Вызов компилятора

- `csc.exe progr.cs`
`/reference:System.Drawing.dll,System.Windows.Forms.dll`
`/target:winexe /out:myprg.exe`
- По умолчанию
 - *mscorlib.dll*
 - *Microsoft.VisualBasic.dll*
- `/reference:<подключаемые библиотеки>`
- `/target:<тип результата>`
 - `exe` – консольное приложение
 - `winexe` – GUI приложение
 - `library` – библиотека классов (dll)
 - `module` – управляемый модуль
- `/out: <имя полученного файла>`