

Уральский федеральный университет
имени первого Президента России Б.Н.Ельцина

Центр инноваций Microsoft

Windows Presentation Foundation

Лекция 1. Введение

Максим Паначёв (max.panachev@live.ru)

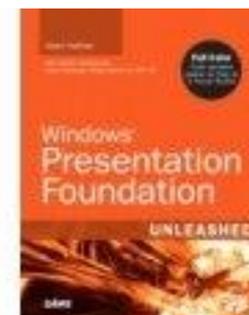
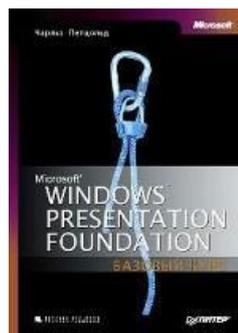
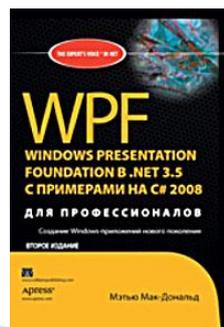
Осень 2012

О курсе

- 12 занятий (3 общих задачи + 1 индивидуальная)
- Дополнительные материалы:
<http://www.techdays.ru>
- Exam 70-502:
TS: Microsoft® .NET Framework 3.5 – Windows®
Presentation Foundation

Книги

- WPF: Windows Presentation Foundation в .NET 3.5 с примерами на С# 2008 для профессионалов (МакДональд М.)
- Microsoft Windows Presentation Foundation. Базовый курс (Петцольд Ч.)
- Основы Windows Presentation Foundation (Андерсон К.)
- Pro WPF in C# 2010 (МакДональд М.)
- WPF 4 Unleashed (Adam Nathan)



Мир Windows UI до WPF

- Все базовые элементы пользовательского интерфейса – сложные бинарные конструкции, вшитые в ядро операционной системы
- Качественное масштабирование при изменении размеров рабочей области – БОЛЬШАЯ проблема!
- Плохая поддержка отрисовки UI через графические ускорители

Идеи WPF

- В каких терминах описывать пользовательский интерфейс?

WinForms – код... много кода!!

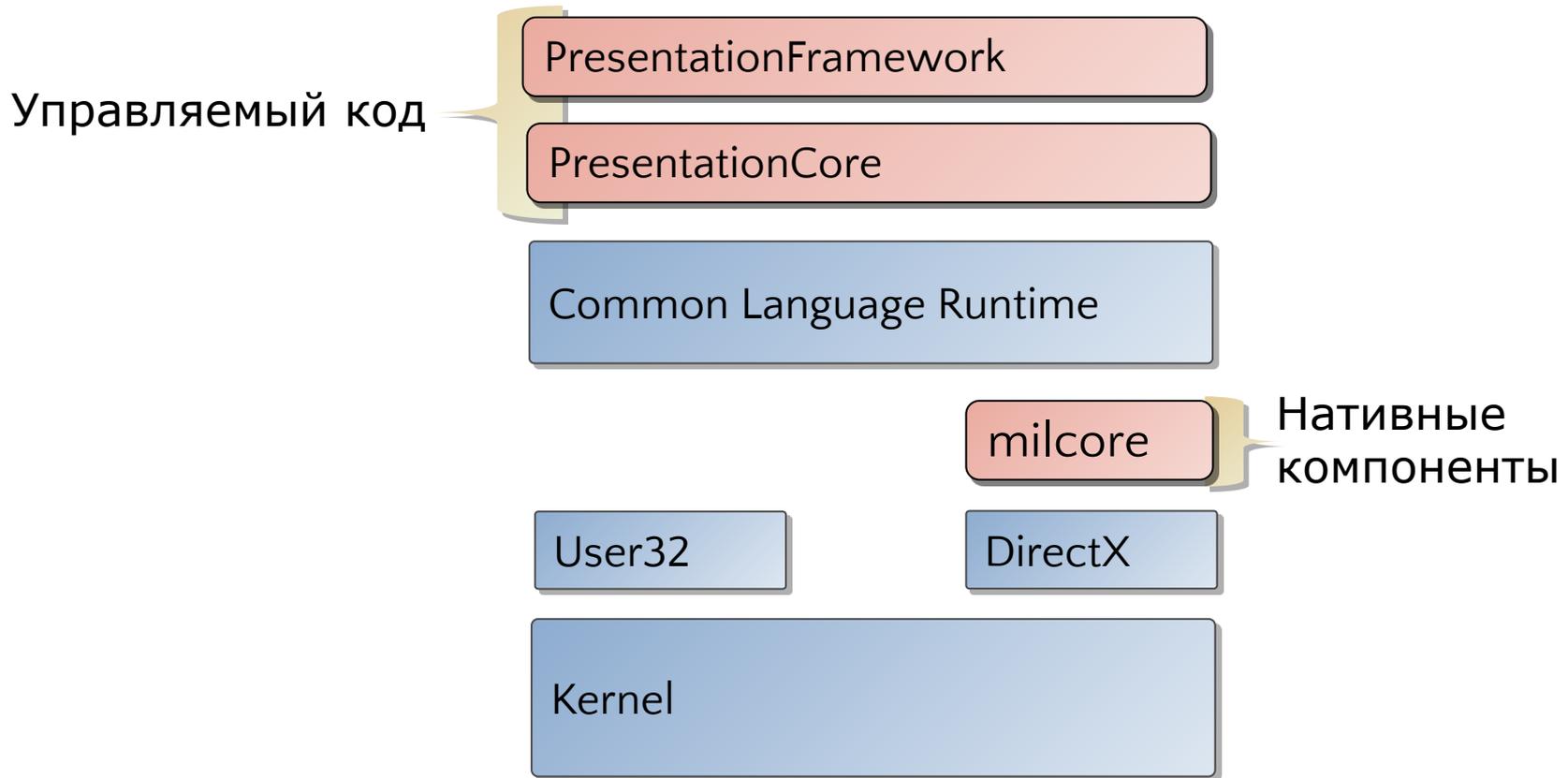
WPF – декларативное описание в виде XML

- Векторная визуализация
- Нативная трансформация UI
- Гибкое масштабирование UI на экран любого разрешения
- Высокая точность – координаты с плавающей точкой

Идеи WPF

- Простота установки – технология ClickOnce
 - Экспозиция приложений через Web
 - Обратная совместимость с существующими платформами
- 

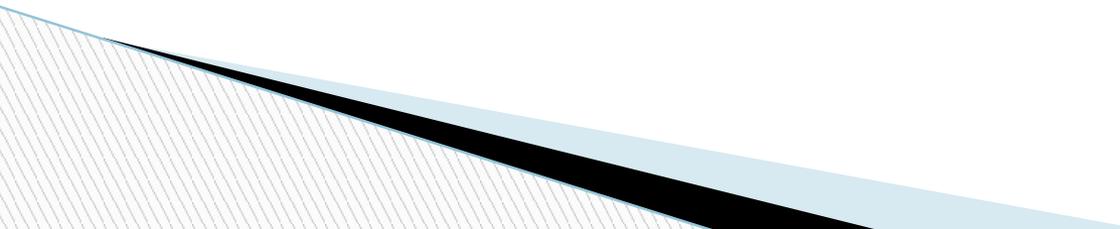
Архитектура WPF



Основные концепции

XAML

eXtensible Application Markup
Language



Основные концепции - XAML

```
<Window ... >  
...  
  <Label>Label</Label>  
  <TextBox>TextBox</TextBox>  
  <RichTextBox ... />  
<RadioButton>RadioButton</RadioButton>  
  <CheckBox>CheckBox</CheckBox>  
  <Button>Button</Button>  
  
</Window>
```

Label

TextBox

Rich**Text**Box

RadioButton

CheckBox

Button

Основные концепции – XAML

□ XAML – это:

- Декларативный язык разметки для пользовательского интерфейса
- Совместная работа дизайнера и разработчика с помощью инструментов

□ XAML компилируется в BAML

- BAML бинарное представление XAML оптимизированное для времени выполнения
- BAML внедряется в ресурсы сборок

□ В XAML может быть подключено CLR пространство имен:

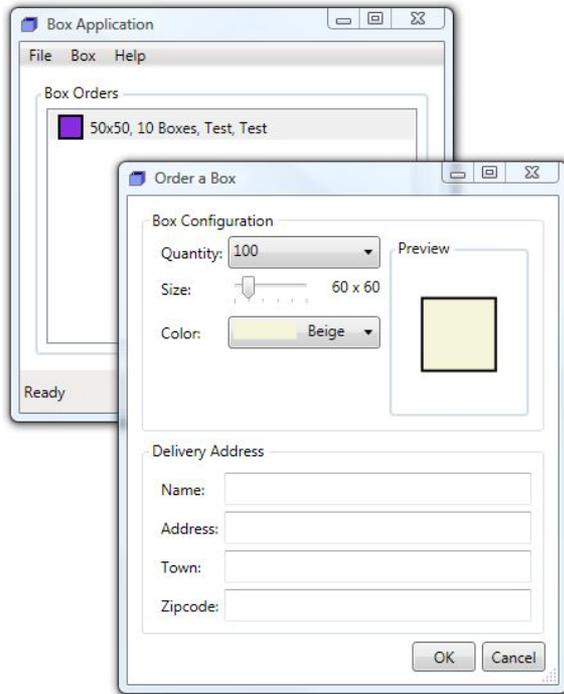
```
xmlns:m="clr-namespace:MyCLRNamespace;assembly=MyAssem"
```

Основные концепции – XAML

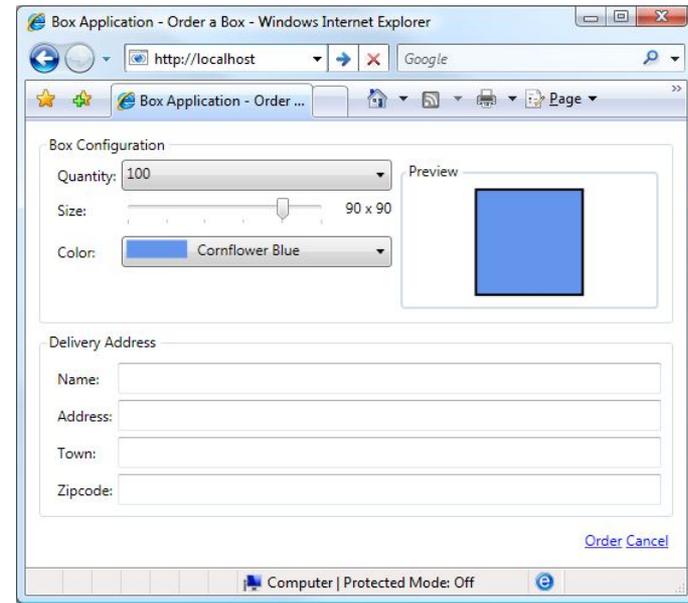
- eXtensible Application Markup Language (XAML)
 - Декларативный язык разметки для пользовательского интерфейса
 - Совместная работа дизайнера и разработчика с помощью инструментов
- XAML компилируется в BAML
 - BAML бинарное представление XAML оптимизированное для времени выполнения
 - BAML внедряется в ресурсы сборок
- В XAML может быть подключено CLR пространство имен:

```
xmlns:m="clr-namespace:MyCLRNamespace;assembly=MyAssem"
```

Основные концепции: исполнение



Stand-Alone
(Классика)



XAML Browser Applications (XBAPs)

Основные концепции: стили / шаблоны

- Базовый класс для пользовательского элемента управления – UserControl
- **Стили** определяют визуализацию и интерактивное поведение элементов
- **Шаблоны** переопределяют визуализацию и интерактивное поведение элементов управления
- **Ресурсы** позволяют повторное использование объектов в элементе управления, контейнере, окне, или всем приложении

Основные концепции: стили / шаблоны



**Элементы
управления
(Controls)**



**Стили
(Styles)**

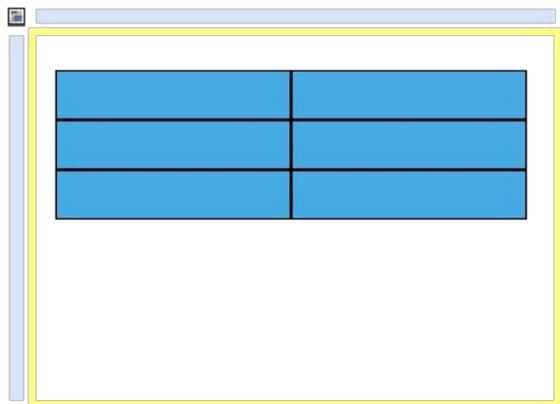


**Шаблоны
(Templates)**

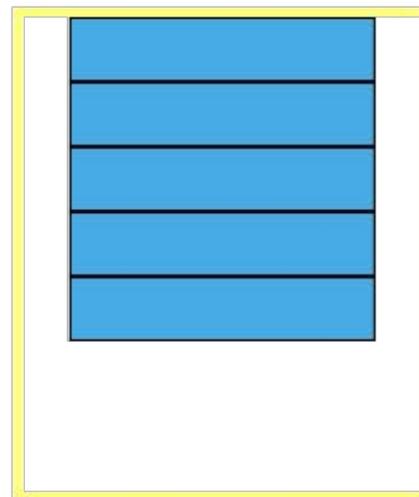
Шаблоны размещения (Layout Templates)

- ▣ **Layout Templates** (шаблоны размещения) определяют размер и расположение элементов пользовательского интерфейса
- ▣ Базовые встроенные шаблоны:
Canvas, DockPanel, StackPanel, Grid, WrapPanel

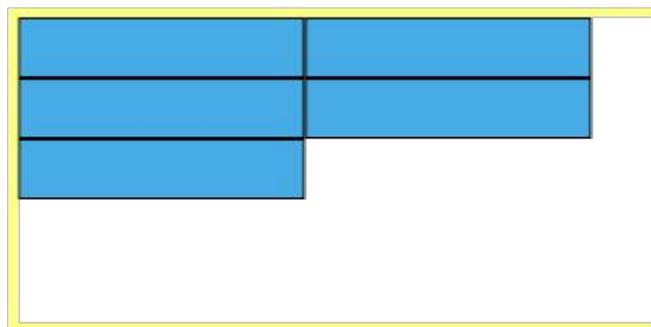
Примеры Layout Templates



Grid



StackPanel



WrapPanel

Шаблоны размещения (Layout Templates)

- Шаблоны размещения:
 - Являются контейнером для других элементов управления
 - Контролируют размер размещённых элементов UI
 - Контролируют положение размещённых элементов UI
- Базовые встроенные шаблоны:
Canvas, DockPanel, StackPanel, Grid, WrapPanel

Анимация в WPF

- В архитектуру WPF встроена система привязки к времени, позволяющая анимировать любое `DependencyProperty` любого `DependencyObject`
- Анимированные свойства меняют свое значение во времени. Это изменяет визуальную сцену.

Dependency Properties

- Свойства зависимостей – особые свойства объектов, используемые при анимации и связывании данных
- Класс, который содержит Dependency Property, обязательно является наследником DependencyObject

```
namespace Test
{
    public class Employee : DependencyObject Регистрация
    {
        public readonly static DependencyProperty NameProperty =
            DependencyProperty.Register("Name",
                typeof(string),
                typeof(Employee));

        public string Name
        {
            get { return (string)GetValue(NameProperty); }
            set
            {
                if (Name == value)
                    return;

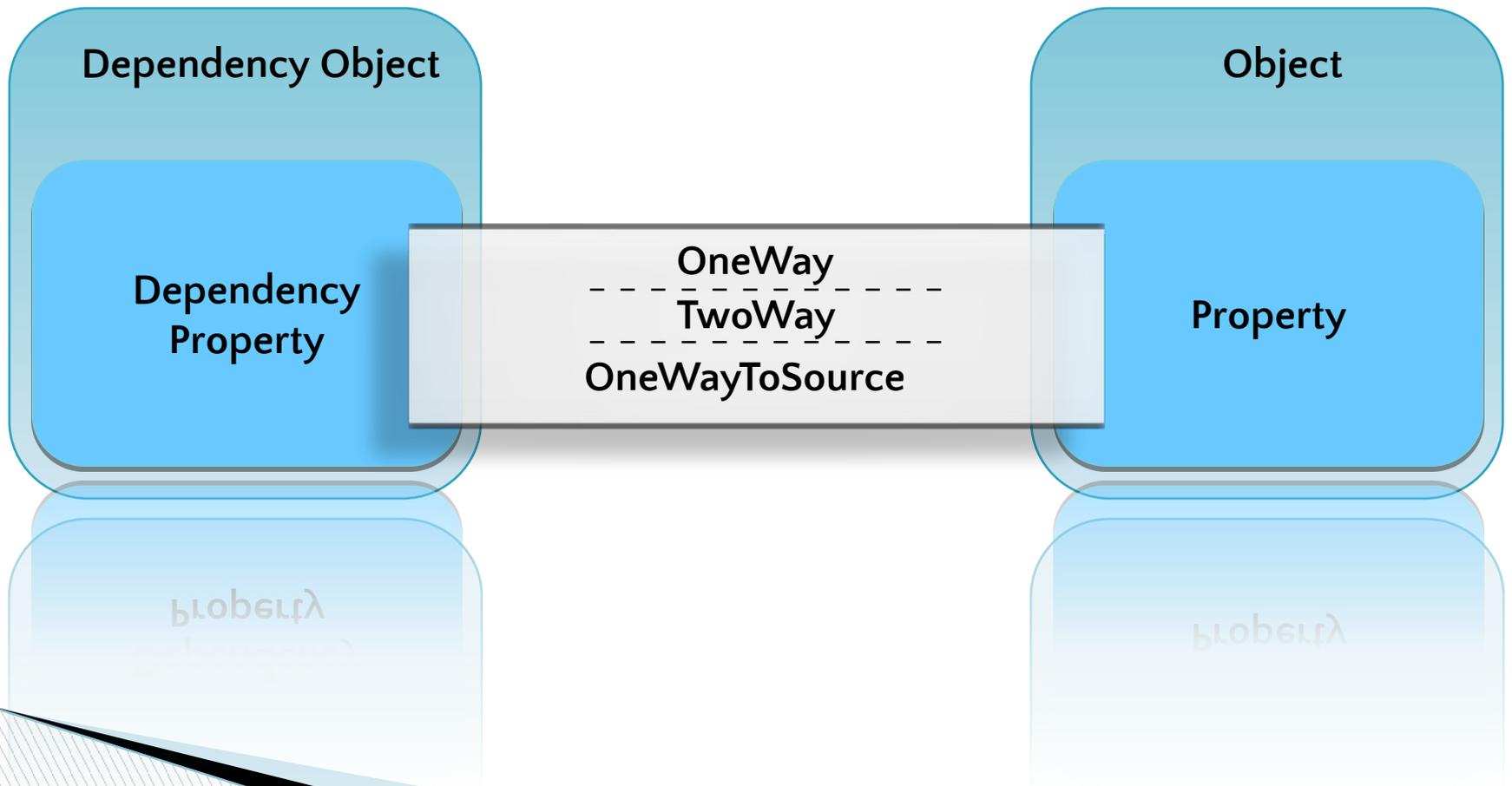
                SetValue(NameProperty, value);
            }
        }
    }
}
```

Фасад

Связывание с данными (Data Binding)

- Data Binding – ассоциирование элементов управления с произвольными данными
 - Когда значение данных меняется, связанные свойства обновляются
 - Когда изменяется связанное свойство, меняются исходные данные
- Можно привязать любое DependencyProperty любого DependencyObject к любому публичному свойству CLR объекта (в том числе и к другому DependencyProperty)
- Более того, элементы управления могут связываться с любыми другими традиционными источниками: XML, ADO.NET, LINQ, ADO Entity Framework

Связывание с данными (Data Binding)



Интегрированная векторная графика

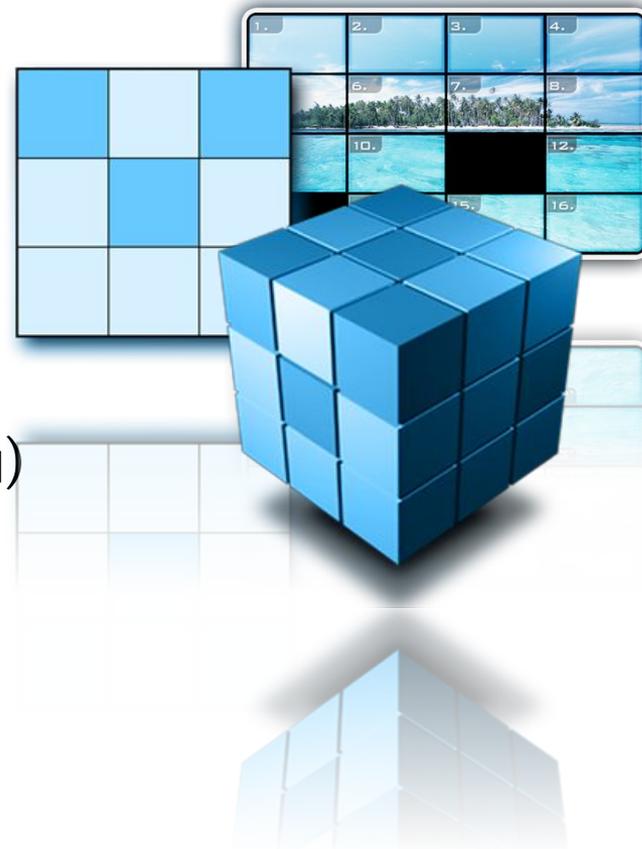
- Современные возможности:
 - 2D и 3D графика
 - Эффекты и поддержка кодеков
 - Аппаратно ускоряемый суб-пиксельный ClearType
 - Видео и аудио
- Графический конвейер: Direct3D аппаратное ускорение
- Готовность к будущему аппаратному обеспечению:
 - Независимость от разрешения
 - Точность с плавающей точкой
 - Расширенные цветовые возможности



2-D, 3-D и изображения

□ Графический API WPF:

- Brushes (Кисти)
- Shapes (Примитивы)
- Imaging (Изображения)
- Geometries (Геометрии)
- Transformations (Трансформации)
- Animations (Анимации)
- Visuals (Визуальные элементы)
- 3-D графика



Развёртывание приложений

□ ClickOnce

- Технология .NET Framework начиная с версии 2.0
- Рекомендуемый метод для WPF
- Приложение «публикуется» на сервер
- Манифест развёртывания определяет параметры
- Проверка версии/обновление происходят каждый раз при запуске приложения

□ MSI установка:

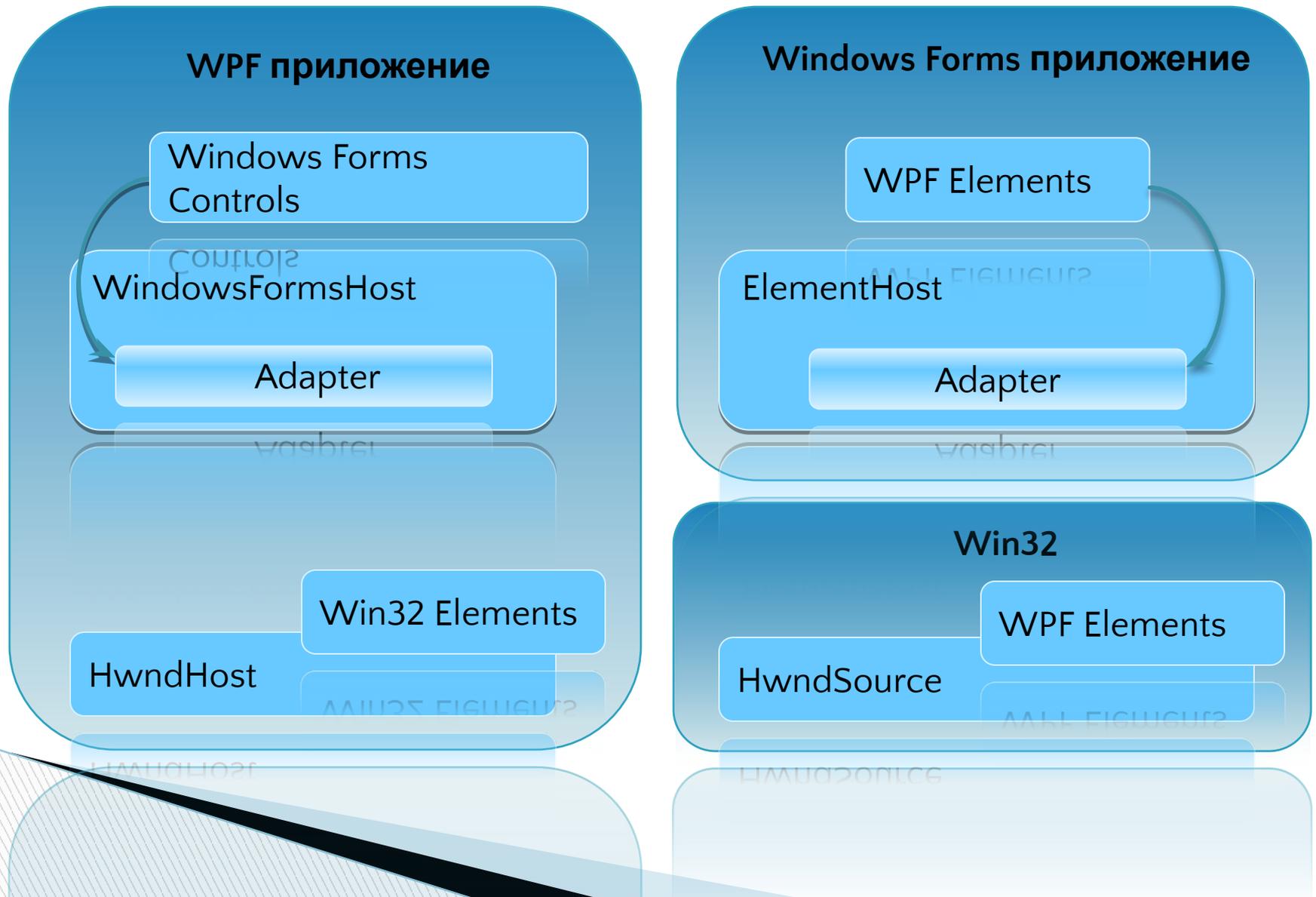
WPF приложения также поддерживают традиционную установку на базе MSI



Обратная совместимость

- WPF может быть «ХОСТОМ»
 - Win32 элемента (HwndHost)
 - WinForms элемента управления (WindowsFormsHost)
 - ActiveX элемента управления (AxHost)
 - WinForm как «ХОСТ» WPF
 - WPF элемента управления (ElementHost)
 - Win32 приложение как «ХОСТ»
 - WPF элемента управления (HwndSource)
- 

Обратная совместимость. Пример



Безопасность



- Full Trust (Полное доверие)
 - Установленные приложения работают в full trust
 - Установленные приложения имеют полный доступ к системным ресурсам
 - Как и Win32 приложения
 - Пользователи устанавливающие приложение через “ClickOnce” получают запрос на авторизацию

- Partial Trust (Частичное доверие)
 - Браузерные приложения
 - Internet Zone permissions
 - Браузерные приложения имеют ограниченный доступ к ресурсам
 - Так как могут нарушить безопасность
 - Браузерные приложения используют механизмы “ClickOnce”, но **без** авторизации.

Основные концепции

- XAML
 - Умная композиция элементов UI
 - Шаблоны и стили элементов UI
 - Data Binding
 - 3D графика
 - Медиа-элементы
 - Анимация
 - Подготовка документов к печати
 - Локализация
 - Обратная совместимость со старыми технологиями разработки UI
- 

Итоги

- WPF – новая мощная платформа для создания графических приложений
- Основой любого WPF приложения является XAML-файл (?!)
- XAML-представление компилируется в BAML-код и помещается в область ресурсов сборки
- Dependency Properties – особые свойства объектов, используемые для анимации и связывания данных
- Для определения вида элемента управления используются стили и шаблоны
- WPF предоставляет двустороннюю совместимость с платформами для создания UI предыдущих поколений

Домашнее задание

1. Создать оконное приложение на WPF без использования XAML-файлов
2. Создать WPF-приложение, хостящее внутри себя WinForms элементы управления

Книги

- WPF: Windows Presentation Foundation в .NET 3.5 с примерами на С# 2008 для профессионалов (МакДональд М.)
- Microsoft Windows Presentation Foundation. Базовый курс (Петцольд Ч.)
- Основы Windows Presentation Foundation (Андерсон К.)
- Pro WPF in C# 2010 (МакДональд М.)
- WPF 4 Unleashed (Adam Nathan)

