# **Практическая работа № 14.** Разработка простого приложения

Объём учебного времени – 2ч

Методические рекомендации

1. Цель работы: продемонстрировать пример создания простейшего приложения WPF.
2. Перечень необходимых средств обучения:
   1. Технические средства обучения:

Компьютер Core i3 3.0, 2 Gb оперативной памяти, винчестер 250 Gb, DVD

* 1. Программное обеспечение:
* программное обеспечение ОС Microsoft Windows 10 Pro (Edu),
* программное обеспечение Microsoft Visual Studio 2017,
* microsoft SQL Server Management
* программное обеспечение Microsoft Office,
* программная платформа .NET Framework 4.6.2

1. Основные теоретические положения:

Графическая система Windows Presentation Foundation предназначена для создания пользовательских интерфейсов, 2D и 3D графики.

Преимущества WPF заключается в том, что 2D графика строится в векторном виде, а это значит, что интерфейсы будут максимально независимы от разрешения экрана и размера окна. Они будут легко масштабироваться без потери качества и быстро работать благодаря максимальному использованию возможностей современных графических ускорителей. WPF объединяет документы, формы и мультимедийное содержание в пакет, состоящий из языка разметки и процедурного языка программирования.

Для создания и инициализации объектов в WPF используется язык разметки XAML - Extensible Application Markup Language (расширяемый язык разметки приложений). XAML использует основные четыре категории элементов: • панели размещения; • элементы управления; • элементы, связанные с документом; • графические фигуры. XAML является диалектом языка XML.

Файл XAML содержит ровно одну корневую вершину и является деревом отображения. На вершине иерархии находится один из контейнерных объектов. Внутри этих объектов располагаются элементы управления и другие контейнеры. В XAML названия элементов чувствительны к регистру и совпадают с именами классов, доступных в кодовой части WPF.



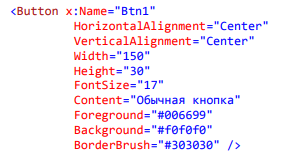
Платформа WPF проектировалась в рамках концепции отделения дизайнерской части пользовательского интерфейса от кодовой части программирования функциональности. Дизайнерская часть проектируется декларативно, с помощью графического дизайнера формы, который в автоматическом режиме генерирует соответствующий синтаксически правильный дескрипторный код на языке XAML.

Объектное описание размещается в пространствах имен, подключаемых в кодовую часть проекта с помощью инструкции using для видимости компилятором. Дескрипторное описание находится в двух пространствах имен: обычном и расширенном. Эти пространства имен подключаются как значения атрибутов xmlns и xmlns:x в корневом дескрипторе проекта

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

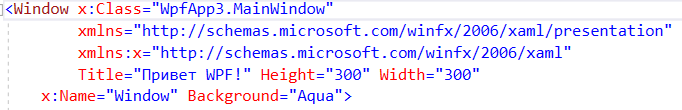
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Используемые URL-адреса не указывают на какой-либо документ или содержимое на веб-сервере, а используются лишь для определения уникальных пространств имен.

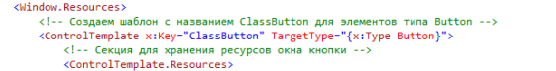


Для описания атрибута используется синтаксис: 

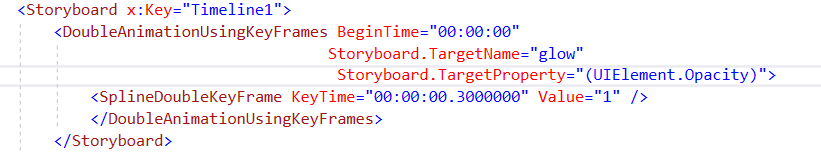
1. Практические указания:
   1. **Простой импорт**
      1. Создайте новый проект Приложения WPF на языке Visual C# с именем WpfApp1 (рис. 1).
      2. Ознакомьтесь с рабочей областью и заготовкой, созданной оболочкой. Оболочка предоставляет возможность графического и дескрипторного способов разработки пользовательского интерфейса, которые являются равнозначными. В обозревателе решений обратите внимание на файлы: дескрипторный MainWindow.xaml и кодовый MainWindow.xaml.cs, которые дополняют друг друга при описании одного и того же содержимого – класса MainWindow (по умолчанию) в пространстве имен WpfApp1.
      3. Отредактируйте настройки окна по представленному примеру:



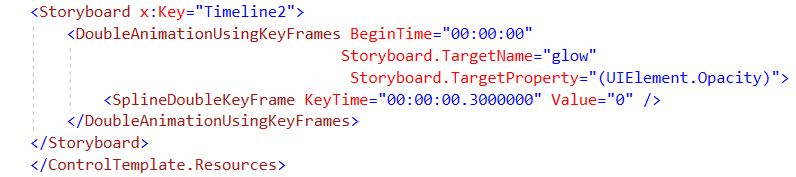
* + 1. Создайте простое изображение мультимедийной кнопки с помощью дескрипторного кода. Обратите внимание, что каждый дескриптор имеет свой тип и атрибуты, уточняющие его настройку. Язык XAML чувствителен к регистру символов, поэтому следуйте указаниям подсказчика IntalliSense. Для этого:
       - опишите секцию Resources для хранения ресурсов окна приложения. Все остальные дескрипторы будут находиться внутри этой секции



* + 1. Далее последовательно продолжайте описывать дескрипторы:
       - секция Storyboard для зажжения кнопки. За установленное время 0,3 секунды кнопка становится непрозрачной:



* + - * секция Storyboard для затухания кнопки. За установленное время 0,3 секунды кнопка становится прозрачной:



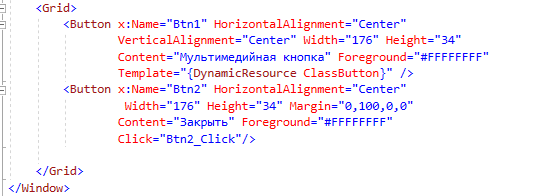
* + 1. В секции для описания кнопки описываем наружную границу белого цвета, внутреннюю границу черного цвета, подсветку кнопки, задаем радиальный градиент со смещением, описываем блик:



* + 1. Далее необходимо задать триггеры, описывающие реакцию на события:



* + 1. И, наконец, создадим экземпляр кнопки посередине окна:



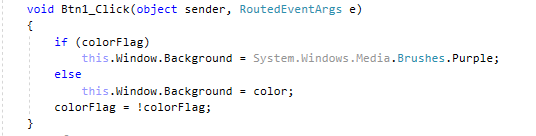
* + 1. Запустите приложение. Посмотрите, результат.
    2. Выберите команду Вид – Другие окна – Структура документа. Сравните представленную структуру с введенным ранее кодом XAML. Нажимая на различны элементы в структуре, обратите внимание, что соответствующий код XAML подсвечивается.
    3. С помощью панели Обозреватель решений откройте файл MainWindow.xaml.cs и дополните его следующим кодом:
       - опишите дополнительные поля для организации смены цвета:



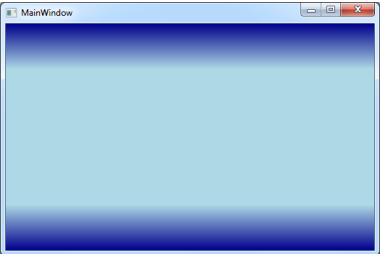
* + - * в конструкторе опишите программно обработчик события щелчка по кнопке Btn1, а также, в переменной color запомните начальный цвет фона окна:



* + - * добавьте метод для изменения цвета фона кнопки:

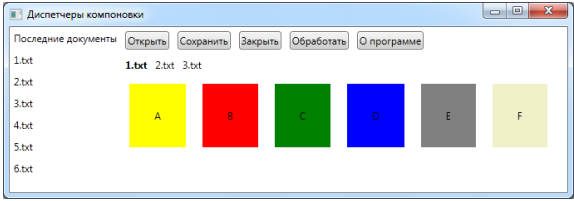


* + 1. Запустите приложение – теперь оно стало реагировать на щелчок мышью по кнопке и менять цвет клиентской области окна.
    2. Добавьте еще одну кнопку, которая будет закрывать форму. Сразу определите обработчик соответствующего события кнопки. В файл MainWindow.xaml.cs допишите обработчик для второй кнопки.
    3. Проверьте работоспособность приложения.
  1. **Задание для самостоятельной работы:**
     1. В XAML-коде для элемента Windows определите линейную градиентную заливку фона в соответствии с рисунком:



Используемые цвета: DarkBlue и LightBlue. Необходимо указать четыре промежуточные точки со смещениями 0, 0.2, 0.8 и 1. Для задания вертикальной заливки необходимо определить атрибуты StartPoint и EndPoint для элемента LinearGradientBrush. Значения этих атрибутов указываются в формате “X,Y”, где X – относительное значение (от 0 до 1) абсциссы точки, Y – относительное значение (от 0 до 1) ординаты точки. Начало координат находится в левом верхнем углу окна. По умолчанию значения атрибутов StartPoint и EndPoint следующие: StartPoint=”0,0” EndPoint=”1,1”

* + 1. Разработать приложение WPF со следующим графическим интерфейсом:



* + 1. Разработать WPF-приложение с меню, панелью инструментов и строкой состояния. С помощью пунктов меню пользователь может изменять цвет фона окна, получить информацию о разработчике, а также закрыть окно. Кнопки панели инструментов дублируют команды меню. При наведении на пункты меню или кнопки панели инструментов в строке состояния отображается информация об этих элементах управления.

1. Контрольные вопросы
   1. Язык XAML.
   2. Пространства имён, элементы, атрибуты.
   3. Перечислите иерархию классов WPF.
   4. Кнопочные, текстовые, линейные элементы
2. Список рекомендуемой литературы:
   1. Агальцов В.П. Базы данных. Распределённые и удалённые базы данных. Учебник М.: ИД «Форум»: ИНФРА-М, 2011. - 272с.
   2. Голицына О.Л. Системы управления базами данных: учеб. пособие.- М.:ФОРУМ: ИНФРА-М,2006.- 432с
   3. Голощапов А.Л. MicrosoftVisualStudio 2010, - СПб.: БХВ-Петербург, 2011. – 544 с.
   4. Грекул В.И. Проектирование информационных систем. Курс лекций. Учебное пособие. М.: Интернет Ун-т Информ. технологий- 2005, 304с.
   5. Зиборов В.В. VisualC# 2012 на примерах. – СПб.: БХВ-Петербург, 2013. – 480 с.
   6. Культин Н.Б. MicrosoftVisualC# в задачах и примерах. – СПб.: БХВ-Петербург, 2013. – 320 с.
   7. Петров В.Н. Избачков Ю.С., Информационные системы. СПб.: Питер, 2006, 656с.
   8. Савицкий Н.М. Технологии организации, хранения и обработки данных: учеб. пособие.- М.:ФОРУМ: ИНФРА-М,2001.- 232с.
   9. Флеонов М.Е. Библия С#. – 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2011. – 560 с.
   10. Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен, Джон Д. Рид, Морган Скиннер.Visual C# 2010: полный курс – М: Вильямс, 2010. – 960 с.
   11. Мак-Дональд, Мэтью. WPF: Windows Presentation Foundation b .NET .3.0 для профессионалов: пер. с англ. / Мэтью Мак-Дональд; пер. Я.П. Волкова [и др.]. – М.; СПб.; Киев: Вильямс, 2008. – 992 с.
   12. Абрамян М.Э. Visual C# на примерах – СПб: БХВ-Петербург, 2008. – 496 с.
   13. Сайт «Национальный открытый университет ИНТУИТ» – http://www.intuit.ru.