

ВВЕДЕНИЕ

Технология *Windows Presentation Foundation* (WPF) была представлена Microsoft в 2006 г. как инструмент создания графического интерфейса для .NET-приложений и в настоящее время рассматривается как наиболее перспективная для дальнейшего развития. WPF представляет собой API-интерфейс для создания настольных графических программ имеющих насыщенный дизайн и интерактивность. В ее основе лежит инфраструктура, основанная на DirectX, — это API-интерфейс графики с аппаратным ускорением, который использует в своей работе ресурсы видеокарты, в отличие от GDI, который использует для графических построений процессор. В связи с этим в WPF становятся доступными также поддержка видеофайлов и трехмерное содержимое.

WPF включает в себя язык разметки *eXtensible Application Markup Language* (XAML) — это расширение языка XML, представляющее собой «декларативный» язык программирования, в отличие от C#, который является «императивным» языком.

Обычно WPF отождествляют с XAML, однако это не так. WPF-проект может быть написан полностью на языке C# либо на любом .NET-совместимом языке, а может и на XAML. Это две совершенно разные технологии, и, как обычно, наибольшая эффективность достигается при совместном использовании языка разметки и кода на алгоритмическом языке.

При создании WPF проекта, а работать мы будем в Visual Studio 2019, мастером-построителем создается cs-файл для частичного (partial) класса окна и xaml-файл для кода разметки. При компиляции, если не вдаваться в подробности, код разметки компилируется отдельно, затем объединяется с кодом, построенным на языке программирования. Если проще, то при помощи декларативной разметки мы строим интерфейс окна, а его интерактивную часть, включающую реакцию на нажатие кнопок и другие события, обрабатываем в императивном коде.

Если посмотреть на код разметки, он очень похож на html-код, однако теги, используемые в XAML-коде, представляют из себя экземпляры классов, которые имеют множество свойств и событий,

позволяющих очень гибко управлять как их внешним видом, так и поведением.

Книга состоит из 15 лекций:

1. **Окна и элементы управления** — создание окна, настройка элементов управления, контейнеры компоновки, ресурсы.

2. **Стили** — создание и наследование стилей, триггеры.

3. **Пространства имен, привязка** — привязка к локальным данным, данным из других сборок, источник и приемник данных. Привязка коллекций.

4. **Элементы управления** — свойства элементов управления содержимым, специальные контейнеры.

5. **Элементы управления списками** — отображение списка, древовидные списки, таблицы.

6. **Элементы управления диапазонов, меню, панель инструментов, строка состояния** — ползунок, индикатор выполнения, стандартные панели.

7. **Графические примитивы** — простейшие графические фигуры, пути. Закраска.

8. **Свойства зависимостей. Модель событий** — автоматическое обновление свойств зависимостей при привязке. Прикрепляемые свойства и события. Маршрутизируемые события.

9. **Команды, класс Application** — стандартные команды, создание команд. Класс Application как источник команд. Свойства и события класса Application.

10. **Шаблоны** — исследование шаблона элемента. Анимация при помощи триггеров. Визуальные состояния.

11. **Окна** — типы окон. Окна стандартных диалогов. Анимация на панели задач.

12. **Страницы** — создание страничного приложения, навигация. WebBrowser.

13. **Документы** — потоковые и фиксированные документы, создание и отображение. Контейнеры потоковых документов.

14. **Работа с данными** — доступ к локальным и внешним данным. Отображение коллекций данных, провайдеры данных. Работа с иерархическими данными.

15. **Многопоточность** — создание многопоточного приложения. Доступ к элементам окна из других потоков.

Технология WPF безгранична, уже на сегодняшний день ее библиотека включает многие тысячи классов. Разумеется, в таком малом объеме мы смогли дать лишь основы подхода к этой технологии. Однако есть с чего начинать для дальнейшего более детального изу-

чения как WPF, так и XAML. Мы опирались на ставшие уже классическими книги [1, 2], а также другие русскоязычные источники [3–6].

Книга предназначена для студентов и аспирантов технических вузов, изучающих информационные технологии, может быть использована для самообразования. Предполагается знание языка программирования C#, хотя бы для консольных приложений. Инструментарий — Visual Studio 2019, который сейчас доступен для свободного скачивания.

Все примеры, приведенные в книге, протестированы и доступны на сайте издательства.