1. Перечислите преимущества и недостатки WPF?

►Веб-подобная модель компоновки

► Богатая модель рисования

► Развитая текстовая модель

► Поддержка аудио и видео

► Приложения на основе страниц

► Декларативный пользовательский интерфейс (XAML) (сочетание)

► Стили и шаблоны. Команды

1. Зачем нужен язык XAML? Каким образом он используется в WPF?

► eXtensible Application Markup Language (XAML)

 Декларативный язык разметки для пользовательского интерфейса

 Совместная работа дизайнера и разработчика с помощью инструментов

 WCF , WF

►Каждый элемент в документе XAML отображается на экземпляр класса .NET.

►код XAML допускает вложение одного элемента внутрь другого ►Свойства каждого класса можно устанавливать через атрибуты или вложенные дискрипторы

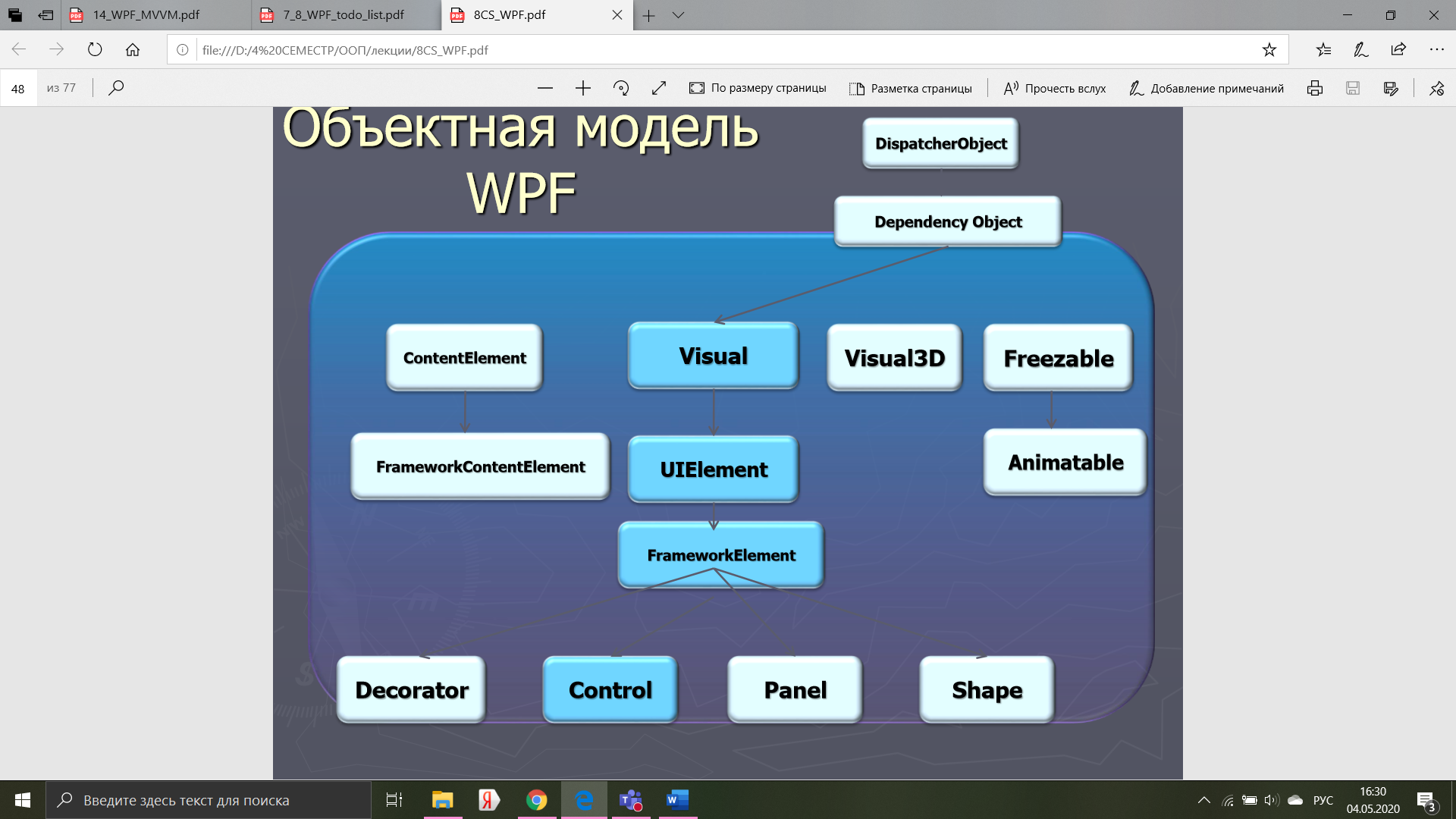
1. Какие бывают контейнеры компоновки?

**Grid**, **UniformGrid**, **StackPanel**, **WrapPanel**, **DockPanel** и **Canvas**. ViewBox

4. Перечислите основные группы элементов управления. Расскажите про объектную модель WPF.

Все элементы управления могут быть условно разделены на несколько подгрупп:

* **Элементы управления содержимым**, например кнопки (Button), метки (Label)
* **Специальные контейнеры**, которые содержат другие элементы, но в отличие от элементов Grid или Canvas не являются контейнерами компоновки - ScrollViewer,GroupBox
* **Декораторы**, чье предназначение создание определенного фона вокруг вложенных элементов, например, Border или Viewbox.
* **Элементы управления списками**, например, ListBox, ComboBox.
* **Текстовые элементы управления**, например, TextBox, RichTextBox.
* **Элементы, основанные на диапазонах значений**, например, ProgressBar, Slider.
* **Элементы для работ с датами**, например, DatePicker и Calendar.
* **Остальные элементы управления**, которые не вошли в предыдущие подгруппы, например, Image.



* 1. Объясните назначение класса DependencyProperty. Зачем нужны свойства зависимостей? Как создать новой свойство зависимости и в каких случаях это необходимо?

Без свойств зависимостей были бы невозможны многие ключевые особенности WPF, как привязка данных, стили, анимация и т.д.

Сначала нужно определить объект, который будет представлять свойство. Это экземпляр класса DependencyProperty.

Чтобы его можно было задействовать, необходимо зарегистрировать свойство зависимости в WPF.

Регистрация свойства зависимости осуществляется в два этапа. Сначала создается объект FrameworkPropertyMetadata, который указывает, какие службы вы хотите использовать со свойством зависимости (например, поддержку привязки данных, анимацию и ведение журнала). Затем свойство регистрируется, для чего вызывается метод DependencyProperty.Register()

* 1. Каким образом осуществляется обработка событий в WPF? Что означает концепция маршрутизированных событий? Перечислите основные группы событий.

События, возникнув на одном элементе, могут обрабатываться на другом. События могут подниматься и опускаться по дереву элементов.

Так, маршрутизируемые события делятся на три вида:

* **Прямые** (direct events) - они возникают и отрабытывают на одном элементе и никуда дальше не передаются. Действуют как обычные события.
* **Поднимающиеся** (bubbling events) - возникают на одном элементе, а потом передаются дальше к родителю - элементу-контейнеру и далее, пока не достигнет наивысшего родителя в дереве элементов.
* **Опускающиеся, туннельные** (tunneling events) - начинает отрабатывать в корневом элементе окна приложения и идет далее по вложенным элементам, пока не достигнет элемента, вызвавшего это событие.

Все маршрутизируемые события используют класс **RoutedEventArgs** (или его наследников),

* 1. Что такое Resource Dictionary?

Чтобы разделить ресурсы между множеством проектов можно создать словарь ресурсов. ***Словарь ресурсов*** представляет собой просто XAML-документ, который всего лишь хранит необходимые ресурсы.

* 1. Что такое стиль и как его создать? В чем преимущество использования стиля?

Стили позволяют определить набор некоторых свойств и их значений, которые потом могут применяться к элементам в xaml.

 Стили помогают создать стилевое единообразие для определенных элементов.

Стиль создается как ресурс с помощью объекта **Style.**

* 1. Что такое Command? Расскажите паттерн Command. Как в WPF используется Command и для чего?

Паттерн "Команда" (Command) позволяет инкапсулировать запрос на выполнение определенного действия в виде отдельного объекта.

в технологии WPF и других технологиях, которые используют XAML и подход MVVM, на командах во многом базируется взаимодействие с пользователем.

Использование команд помогает нам сократить объем кода и использовать одну и ту же команду для нескольких элементов управления в различных местах программы. Таким образом, команды позволяют абстрагировать набор действий от конкретных событий конкретных элементов.