**1. Преимущества и недостатки WPF:**

**Преимущества:**

1. Мощная система разметки XAML, которая обеспечивает разделение дизайна и логики приложения. 2. Поддержка векторной и анимированной графики. 3.Использование стилей, шаблонов и ресурсов. 4. Интеграция с другими технологиями и инструментами .NET, такими как ASP.NET.

**Недостатки:**

1. Использование ресурсов компьютера из-за большего потребления памяти и процессорного времени. 2. Ограничения в производительности при работе с анимацией и графикой, особенно на старых или менее мощных устройствах. 3. Сложность в изучении и понимании для новичков из-за множества концепций и абстракций.

2. Язык XAML (eXtensible Application Markup Language) - для описания пользовательского интерфейса приложения в виде XML-подобного кода. Он разделяет дизайн и логику приложения.

3. Контейнеры компоновки в WPF:

StackPanel Grid Canvas DockPanel WrapPanel

4. Основные группы элементов управления:

Элементы управления контента (например, TextBlock, Button, TextBox)

Элементы управления списками (например, ListBox, ListView)

Элементы управления данных (например, DataGrid, DataTemplate)

Элементы управления макетом (например, StackPanel, Grid)

Элементы управления ввода (например, CheckBox, RadioButton)

Объектная модель WPF - иерархия классов, которая описывает ЭУ.

5. Класс DependencyProperty - для создания свойств, которые могут быть привязаны, анимированы и использованы в шаблонах в WPF. Свойства зависимостей - для привязки данных, анимации и стилей. Новое свойство зависимости можно создать, определив его как статическое поле типа DependencyProperty в классе зависимого объекта. Это необходимо в случаях, когда требуется расширить стандартные возможности свойств.

6. Обработка событий в WPF осуществляется через привязку обработчиков событий к ЭУ. Маршрутизированные события - механизм, который позволяет событиям передаваться вверх или вниз по иерархии ЭУ. Основные группы событий: прямые, поднимающиеся, опускающиеся.

7. Resource Dictionary - это специальный тип XAML-файла, который содержит ресурсы, такие как стили, шаблоны и ресурсы данных. Он позволяет группировать ресурсы в одном месте для повторного использования в различных частях приложения.

8. Стиль - набор свойств, который можно применить к нескольким элементам интерфейса. Преимущество: он позволяет легко изменять внешний вид группы элементов интерфейса.

9. Command - объект, который инкапсулирует логику действия, которое может быть выполнено в ответ на пользовательское действие. Паттерн Command - для отделения логики выполнения действия от его вызывающего объекта. В WPF - для связывания событий интерфейса с методами.