1) Что такое MVC, MVP, MVVM?

архитектурные шаблоны, которые используются для разделения компонентов пользовательского интерфейса и бизнес-логики в приложениях.

MVC:

* Model (Модель) представляет данные и бизнес-логику приложения.
* View (Представление) отображает данные пользователю и реагирует на его взаимодействие.
* Controller (Контроллер) обрабатывает входящие запросы от пользователя, взаимодействует с моделью и обновляет представление.

MVP:

* Model (Модель) содержит данные и бизнес-логику приложения.
* View (Представление) отвечает за отображение данных и обработку пользовательского ввода.
* Presenter (Презентер) является посредником между моделью и представлением, обрабатывает события от представления, обновляет модель и обновляет представление в соответствии с изменениями в модели.

MVVM:

* Model (Модель) представляет данные.
* View (Представление) отображает данные и реагирует на пользовательский ввод.
* ViewModel (Модель представления) связывает представление и модель, предоставляет данные и команды для представления, а также обрабатывает пользовательский ввод и обновляет модель.

6) Чем отличаются команды от событий?

Команда обычно связана с конкретным объектом или элементом пользовательского интерфейса.

Обработчик команды определяет логику выполнения команды.

Событие может быть инициировано объектом или системой и передает информацию о произошедшем событии.

Событие может быть подписано (или обработано) различными обработчиками, которые выполняют определенные действия в ответ на возникшее событие.

7)

Light Toolkit - это набор инструментов и библиотек, предназначенных для создания пользовательского интерфейса (UI) в приложениях

Catel - это фреймворк для разработки приложений на языке C# с использованием паттернов проектирования Model-View-ViewModel (MVVM) и Inversion of Control (IoC). Он предоставляет набор инструментов и функций, упрощающих разработку сложных приложений.

Prism - это фреймворк для разработки гибких и масштабируемых приложений на платформе .NET, основанных на паттернах проектирования Model-View-ViewModel (MVVM) и инверсии управления (Inversion of Control - IoC). Он предоставляет множество функций, таких как навигация, модульность, управление состоянием и взаимодействие между компонентами приложения.