|  |
| --- |
| 1. Что такое MVC, MVP, MVVM? |
|  |  |
|  | **MVC-MVC** — это шаблон программирования, который позволяет разделить логику приложения на три части: |
|  | Model (модель). Получает данные от контроллера, выполняет необходимые операции и передаёт их в вид. |
|  | View (вид или представление). Получает данные от модели и выводит их для пользователя. |
|  | Controller (контроллер). Обрабатывает действия пользователя, проверяет полученные данные и передаёт их модели. |
|  |  |
|  | **MVP**-Model-View-Presenter — довольно известный шаблон проектирования. С первого взгляда все выглядит просто: есть Модель (Model), |
|  | которая содержит всю бизнес-логику экрана; Вид/Представление (View), |
|  | который знает, как отобразить те или иные данные; Представитель (Presenter), |
|  | который является связующий звеном — реагирует на действия пользователя во View, изменяя Model, и наоборот. |
|  |  |
|  | **MVVM**-озволяет отделить логику приложения от визуальной части (представления). |
|  | Данный паттерн является архитектурным, то есть он задает общую архитектуру приложения. |
|  | MVVM состоит из трех компонентов: модели (Model), модели представления (ViewModel) и представления (View). |
|  | Модель описывает используемые в приложении данные. |
|  | View или представление определяет визуальный интерфейс, через который пользователь взаимодействует с приложением. |
|  | ViewModel или модель представления связывает модель и представление через механизм привязки данных. |
|  |  |
|  | 2. Объясните из каких компонентов состоит паттерн MVVM? |
|  |  |
|  | MVVM состоит из трех компонентов: модели (Model), модели представления (ViewModel) и представления (View). |
|  |  |
|  | View <--> ViewModel (привязка данных, команды) |
|  | ViewModel --> Model(обновление модели) |
|  | ViewModel <-- Model(получение данных) |
|  |  |
|  |  |
|  | 3.Каким образом паттерн можно реализовать в WPF? |
|  |  |
|  | \*Построение модели – EF |
|  | \*Создание репозитория |
|  | \*Реализация репозитория |
|  |  |
|  | 4.Объясните назначение интерфейса INotifyPropertyChanged? |
|  |  |
|  | Если в модели изменяются значения свойств, |
|  | при реализации моделью интерфейса INotifyPropertyChanged автоматически идет изменение отображаемых данных в представлении, |
|  | хотя напрямую модель и представление не связаны. |
|  |  |
|  | 5.Зачем нужен интерфейс ICommand и как паттерн MVVM использует |
|  | команды в WPF-приложениях. |
|  |  |
|  | В MVVM не пишутся обработчики событий. |
|  | Функции, которые нужно выполнять контролам, пишутся во ViewModel и биндятся к контролам точно так же, как поля. |
|  | Только используется механизм команд. |
|  | Команда должна представлять из себя экземпляр класса, реализующего интерфейс ICommand. |
|  | Для привязки команды кнопки необходимо привязать свойство, которое является реализацией ICommand. |
|  |  |
|  | 6.Чем отличаются команды от событий? |
|  |  |
|  | Команды - это более стандартный способ интеграции событий. они могут быть более полезными, чем события, |
|  | потому что с их помощью вы можете определить одну задачу (команду) и использовать ее из разных мест. |
|  | например, вы можете определить команду сохранения и одновременно использовать пункт меню, пункт контекстного меню и кнопку. |
|  | таким образом вы можете центрировать задачи. также команды поддерживают привязку данных, которая является очень мощной функцией. |
|  |  |
|  | 7.Расскажите о назначении и об основных возможностях следующих |
|  | библиотек и фреймворков: Light Toolkit, Catel и Prism. |
|  |  |
|  | Light Toolkit добавляет в VS новый тип проекта – таким образом не нужно создавать стр-ру проекта и все сделано за нас |
|  | Это кросс-платформенная библиотека с открытым исходным кодом для построения граф.интерф.юзера. |
|  | PRISM позволяет проектировать и писать приложения со слабо связанными компонентами, кот.могут независимо развиваться и потом объединяться в одно целое без особых усилий |
|  | CATEL – платформа для разработки приложений. Цель – предоставить полный набор функций для приложений, написанных на любой .NET технологии. |