

## **Контрольные вопросы и ответы к лекции №16**

### **1. Что такое MVC (Model-View-Controller)?**

MVC (Model-View-Controller) - это архитектурный паттерн, который используется в разработке программного обеспечения для разделения приложения на три основных компонента: модель (Model), представление (View) и контроллер (Controller). Каждый из этих компонентов отвечает за определенную задачу в приложении.

### **2. Какой цели служит компонента модель в архитектуре MVC?**

Компонента модель в архитектуре MVC служит для управления данными и бизнес-логикой приложения. Модель отвечает за получение, хранение и обработку данных, а также за уведомление представления о любых изменениях данных.

### **3. Какой цели служит компонента вид в архитектуре MVC?**

Компонента вид в архитектуре MVC служит для отображения данных пользователю. Представление отвечает за отображение данных, полученных из модели, и за предоставление пользователю интерфейса для взаимодействия с приложением.

### **4. Какой цели служит компонента контроллер в архитектуре MVC?**

Компонента контроллер в архитектуре MVC служит для управления потоком данных между моделью и представлением. Контроллер получает запросы от пользователя через представление, обрабатывает эти запросы и обновляет модель, а затем обновляет представление на основе изменений в модели.

### **5. Какие шаги выделяют при проектировании с использованием архитектуры MVC?**

При проектировании с использованием архитектуры MVC выделяют следующие шаги:

- Определение модели данных и бизнес-логики приложения.
- Определение пользовательского интерфейса и представления данных.
- Определение контроллера, который будет управлять потоком данных между моделью и представлением.
- Разработка и тестирование каждой компоненты по отдельности.
- Интеграция всех компонентов и тестирование приложения в целом.

#### **6. Какие основные шаблоны проектирования используются при реализации архитектуры MVC?**

Основные шаблоны проектирования, которые используются при реализации архитектуры MVC, включают:

- Наблюдатель (Observer) - используется для уведомления представления об изменениях в модели.
- Фабричный метод (Factory Method) - используется для создания объектов модели и представления.
- Стратегия (Strategy) - используется для определения различных стратегий обработки данных в модели.
- Адаптер (Adapter) - используется для связывания представления и модели, чтобы они могли взаимодействовать друг с другом.