**1. Что такое фреймворк и зачем они нужны**

Фреймворк – это набор инструментов, библиотек и руководств по структурированию и разработке программного обеспечения. Фреймворки предоставляют готовые компоненты и решения для общих задач в программировании, что позволяет разработчикам сосредоточиться на уникальных аспектах своих проектов, сокращая время разработки и повышая качество кода.

Сравнение фреймворков для PHP: Symfony, Laravel, Yii2

1. Symfony:

- Основан на компонентах, что позволяет использовать его части в других проектах.

- Сильная стандартизация и структурирование кода.

- Хорошо подходит для крупных и долгосрочных проектов благодаря строгой архитектуре и масштабируемости.

2. Laravel:

- Очень популярен среди разработчиков благодаря своей простоте и интуитивно понятному синтаксису.

- Включает множество функций "из коробки", таких как аутентификация, авторизация, и система миграции баз данных.

- Имеет обширное сообщество и множество пакетов расширений.

3. Yii2:

- Отличается высокой производительностью и подходит для проектов с большим объемом данных.

- Прост в установке и настройке.

- Предоставляет мощные инструменты для генерации кода, что ускоряет разработку.

**2. Шаблон MVC (Model-View-Controller)**

Сфера применения и принципы:

- MVC широко используется в веб-разработке для разделения данных приложения (модель), пользовательского интерфейса (вид) и логики управления (контроллер).

- Модель управляет данными и бизнес-логикой.

- Вид отображает данные (модель) и отправляет пользовательские команды (например, клик по кнопке) контроллеру.

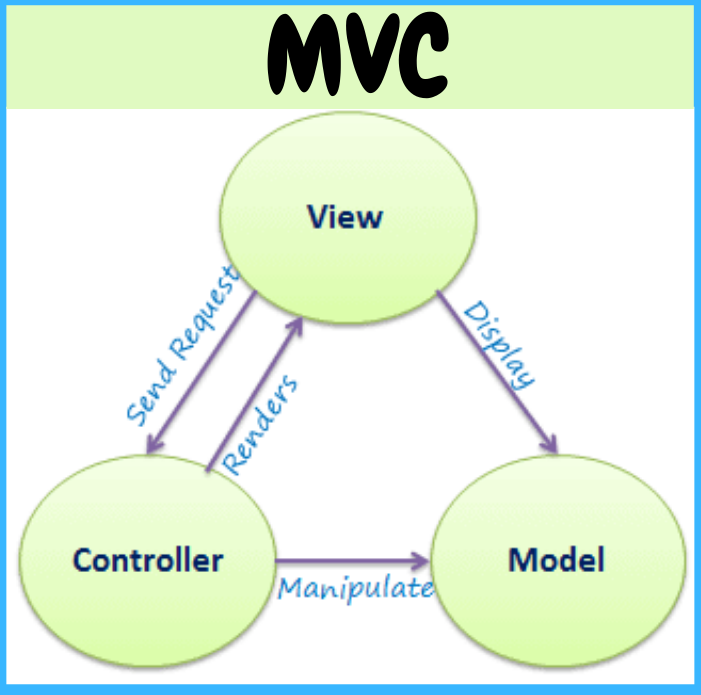
- Контроллер обрабатывает входные данные (от пользователя или интерфейса) и обновляет модель.

Сравнение с MVVM (Model-View-ViewModel):

- MVVM предназначен для упрощения программной логики путем создания связующего слоя (ViewModel) между видом и моделью, что позволяет облегчить управление зависимостями и динамически обновлять интерфейс пользователя.

- MVVM чаще используется в приложениях с богатым клиентским интерфейсом, где требуется сложная логика динамического взаимодействия, например в WPF или Xamarin.

Диаграмма MVC:



**3. Шаблон «Репозиторий» vs Active Record**

Репозиторий:

- Представляет собой слой абстракции, который позволяет делать запросы к данным, не завися от конкретной структуры базы данных.

- Применяется для

разделения логики бизнес-слоя и доступа к данным, что упрощает тестирование и модификацию.

Active Record:

- Каждый объект модели в Active Record также отвечает за сохранение себя в базу данных.

- Прост в реализации, но может привести к смешиванию бизнес-логики и логики доступа к данным.

**4. Dependency Injection в Symfony**

Принципы:

- Dependency Injection (DI) – это метод передачи зависимостей объектам. Вместо того, чтобы объекты самостоятельно создавали нужные им зависимости или запрашивали их, они получают все необходимое извне, что делает систему более гибкой и облегчает тестирование.

Реализация в Symfony:

- Symfony использует контейнер служб для управления зависимостями. Компоненты и службы описываются в конфигурационных файлах, и контейнер автоматически их инстанцирует и управляет их жизненным циклом.

**5. События и обработка событий в Symfony**

Определения:

- Событие – это сигнал о том, что что-то произошло. В программировании это часто связано с действиями пользователя или изменениями в системе.

- Обработчик события – это функция или метод, вызываемый в ответ на событие.

Обработка событий в Symfony:

- Symfony предоставляет компонент EventDispatcher для управления событиями. Разработчики могут добавлять слушателей к определенным событиям, и когда событие происходит, EventDispatcher вызывает все зарегистрированные обработчики.