1. Класс Dog   
   Этот код определяет класс Dog, который наследует от класса Animal. Он также помечен аннотацией @Component, что делает его бином Spring Framework и позволяет Spring автоматически управлять его жизненным циклом. Метод toString() переопределен для возвращения строки "I'm a Dog", что позволяет возвращать эту строку при вызове метода toString() для объектов Dog.
2. Class AnimalsCage Аннотация @Component: Эта аннотация указывает Spring, что класс AnimalsCage является компонентом, который должен быть зарегистрирован в контексте Spring. Это означает, что Spring автоматически создаст экземпляр этого класса и управит его жизненным циклом.

Поля класса:

private final Animal animal; – Это поле хранит объект типа Animal. Оно помечено как final, что означает, что оно должно быть инициализировано только один раз и не может быть изменено после этого.

private final Timer timer; – Это поле хранит объект типа Timer, также помеченный как final.

Конструктор класса:

public AnimalsCage(@Qualifier("dog") Animal animal, Timer timer) – Конструктор принимает два параметра: @Qualifier("dog") Animal animal – Использование аннотации @Qualifier позволяет Spring выбрать конкретный бин из множества возможных, если есть несколько реализаций интерфейса Animal. В данном случае выбирается бин с именем dog.

Timer timer – Параметр конструктора, представляющий таймер, который будет инжектироваться в этот класс.

Методы класса:

public Timer getTimer() – Метод доступа (геттер) для получения значения поля timer.

public void whatAnimalSay() – Этот метод выводит на консоль информацию о том, что говорит животное, а также текущее время. Он вызывает метод toString() у объекта animal для получения строкового представления животного и метод getTime() у объекта timer для получения текущего времени.

1. Class AppConfig   
   @Configuration:Это аннотация, которая указывает, что класс AppConfig является источником конфигурации для Spring. Spring будет использовать этот класс для создания и настройки бинов (объектов) приложения.

@ComponentScan(basePackages = "app"): Аннотация @ComponentScan говорит Spring, что он должен сканировать пакеты, указанные в basePackages, для поиска компонентов, таких как бины, сервисы и репозитории. В данном случае, Spring будет искать компоненты в пакете app и его подпакетах.

public class AppConfig: Это класс конфигурации, который используется для настройки Spring контейнера. Поскольку он помечен аннотацией @Configuration, Spring знает, что этот класс содержит методы для создания и настройки бинов.

Таким образом, этот класс говорит Spring, что он должен искать компоненты в пакете app, и настроить их для использования в приложении.