* Spring is an open-source Java application framework that simplifies the development of enterprise-level Java applications. It offers comprehensive infrastructure support and features like dependency injection, aspect-oriented programming, transaction management, and more. The framework aims to address common challenges in enterprise application development such as configuration overhead, boilerplate code, and coupling.
* Spring Boot is an add-on to the Spring framework that aims to simplify the process of creating new Spring applications. It follows a convention-over-configuration approach, which means that it comes with default configurations and settings to quickly get the application up and running with minimal manual configuration. Additionally, Spring Boot is highly opinionated, meaning that it reduces the need for developers to make decisions about frameworks and libraries, allowing them to focus primarily on the application logic.
* Spring Boot is a framework that is developed on top of the Spring platform. Spring platform is a comprehensive set of tools and features that are used for building enterprise applications. However, Spring Boot simplifies and streamlines the process of configuring and setting up Spring-based applications. Essentially, Spring Boot is an extension of the Spring platform, designed to offer additional capabilities that enhance developer productivity.
* The Spring Framework is the foundation of the entire Spring ecosystem, offering essential features and components for creating enterprise Java applications. Spring Boot, Spring MVC, Spring Data, and other Spring projects are developed on top of the Spring Framework, using its capabilities to provide more specialized functionalities for different aspects of application development.
* Dependency Injection is a software design pattern that helps manage dependencies between objects in a software application. In Spring, the Dependency Injection technique is achieved through the Inversion of Control (IoC) container. This IoC container manages the creation and wiring of objects. As a result, developers can declare dependencies in configuration files or by using annotations. At runtime, the IoC container then injects the required dependencies into the dependent objects.
* Inversion of Control (IoC) is a fundamental principle that enables the control of object creation and lifecycle to be inverted from the application code to a container or framework. In the context of Spring, IoC is implemented through the Spring IoC container, which handles the instantiation, configuration, and lifecycle of application objects. By following IoC, developers can focus on defining components and their relationships, while the Spring container takes care of instantiating and managing these components, promoting loose coupling and easier maintenance.