



SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Coimbatore - 641035.



*Accredited by NBA - AICTE and Accredited by NAAC - UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai*

Department of Computer Applications

Course Code: 23CAT606

Course Name: Java Programming

Unit : V

Topic : Inner Beans, Aliases in Bean,
Bean Scopes





Inner Beans

1

Nested Dependency

Inner beans are defined as a property of another bean, creating a nested dependency structure.

2

Scoping and Lifecycle

Inner beans have their own scoping and lifecycle, independent of the containing bean.

3

Encapsulation

Inner beans help encapsulate complex object graphs within a single bean definition.



Bean Aliases

Alternative Names

Aliases provide alternative names for a bean, allowing for more flexibility in how the bean is referenced.

Locating Beans

Aliases make it easier to locate and access beans within the application context.

Decoupling

Aliases decouple the bean name from its implementation, promoting modular and maintainable code.



Bean Scopes

Singleton

The default scope, where a single instance of the bean is shared across the entire application.

Prototype

A new instance of the bean is created each time it is requested from the container.

Request

A new instance of the bean is created for each HTTP request in a web-aware Spring application.

Session

A new instance of the bean is created for each HTTP session in a web-aware Spring application.



Singleton Scope

1

Application Startup

The Singleton bean is instantiated and stored in the application context.

2

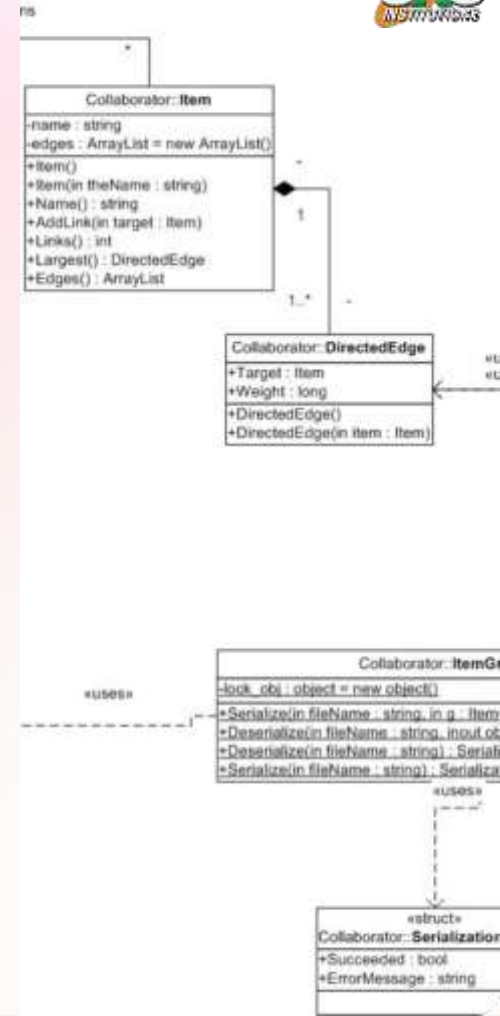
Bean Injection

Whenever the bean is requested, the same shared instance is returned.

3

Lifecycle Management

The Singleton bean's lifecycle is managed by the Spring container.





Prototype Scope

Bean Request

Each time the bean is requested, a new instance is created and returned.

1

2

3

Dependency Injection

Prototype beans are often used for stateful objects that require new instances.

Lifecycle Management

The Spring container does not manage the lifecycle of Prototype beans.



Request and Session Scopes



Request Scope

A new instance of the bean is created for each HTTP request in a web application.



Session Scope

A new instance of the bean is created for each HTTP session in a web application.



Lifecycle Management

The Spring container manages the lifecycle of Request and Session scoped beans.



Inner Beans in Spring

Nested Beans

Inner beans, also known as nested beans, are beans that are defined within the scope of another bean. They are typically used to encapsulate complex dependencies or configurations within a parent bean.

Configuration Simplicity

Inner beans allow for a more concise and organized configuration, as the nested bean's properties can be defined directly within the parent bean's definition.

Scoping and Lifecycle

Inner beans are scoped to the lifecycle of their parent bean, and they cannot be referenced independently outside of the parent bean.



Aliases for Beans in Spring

Multiple Names

Aliases allow a single bean to be referenced by multiple names, providing flexibility and alternative ways to access the same bean.

Flexibility

Aliases can be used to maintain backwards compatibility, provide alternative naming conventions, or to create more descriptive names for the same bean.

Configuration

Aliases can be defined in the bean configuration, either through XML, Java-based, or annotation-based configuration.



Bean Scopes in Spring

Singleton

The default scope, where a single instance of the bean is shared across the entire application.



Prototype

A new instance of the bean is created each time it is requested from the container.

Request, Session, Application, WebSocket

Scopes specific to web applications, where the bean's lifecycle is bound to the corresponding HTTP request, session, application, or

WebSocket.





Sample Program: Inner Beans

1

Nested Configuration

Defining an inner bean for a dependency within the parent bean's configuration.

2

Encapsulation

Encapsulating complex configurations and dependencies within the parent bean.

3

Lifecycle Management

Inner beans inheriting the lifecycle of the parent bean, simplifying bean management.





Sample Program: Aliases



Flexible Lookup

Accessing the same bean using different names or aliases.



Backwards Compatibility

Maintaining existing references to the bean by providing aliases.



Descriptive Naming

Using aliases to provide more meaningful names for the same bean.



Sample Program: Bean Scopes

Singleton

A single instance shared across the application.

Prototype

A new instance created for each request.

Request

Scoped to the current HTTP request.

Session

Scoped to the current HTTP session.

Application

Scoped to the ServletContext.

WebSocket

Scoped to the current WebSocket session.