Advanced Programming

Week 14

Java Beans

- Introduction to Java Beans
- Advantages of Java Beans
- Steps creating Java Beans

Tilahun Melak(PhD)

November, 2025

Objectives

At the end of this lecture, students will be able to:

- Define purpose and architecture of JavaBeans.
- Explain properties, events, and persistence.
- Describe customization and component reuse.
- Outline Bean creation, testing, and packaging.
- Explore advanced features, security, and IDE integration.

Introduction to JavaBeans

- JavaBeans is a reusable component model for Java software.
- It enables developers to build applications by assembling pre-built components.
- Introduced to support visual programming tools.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Benefits of JavaBeans

- Traditional Java classes lacked a standardized way for reuse in visual tools.
- Need for consistent naming conventions and property exposure.
- JavaBeans solves this through introspection, properties, and events.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Features of JavaBeans

- Reusability and modularity.
- Platform independence.
- Support for introspection, properties, events, and persistence.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

JavaBeans Architecture

- Beans are Java classes that adhere to specific conventions.
- Introspection allows tools to analyze bean capabilities.
- Event handling supports component interaction.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

JavaBeans Design Principles

- Encapsulation of behavior and state.
- Customizable properties.
- Loose coupling between components through events.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Introduction to Properties

- Properties represent attributes of a Bean.
- Accessed via getter and setter methods.
- Enable data encapsulation and customization.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Simple and Indexed Properties

- Simple properties hold single values.
- Indexed properties store arrays or lists.
- Naming convention: getProperty, setProperty.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Bound and Constrained Properties

- Bound: notifies listeners when property changes.
- Constrained: allows listener to veto a change.
- Support dynamic component interaction.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Event Model in JavaBeans

- Uses Java's delegation event model.
- Event sources fire events to registered listeners.
- Promotes decoupling between components.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Creating Event Listeners

- Implement event listener interfaces.
- Register listeners with event sources.
- Handle events through callback methods.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Introspection Overview

- Enables tools to discover a Bean's capabilities.
- Performed via reflection or BeanInfo classes.
- Determines properties, events, and methods.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

BeanInfo Interface

- BeanInfo provides explicit metadata about a Bean.
- Used to override default introspection behavior.
- Helps visual tools customize Bean display.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Persistence in JavaBeans

- Beans use serialization to persist state.
- Allows saving and restoring configurations.
- Supports long-term data storage.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Serialization Example

- Use java.io.Serializable interface.
- Save object state to a stream.
- Reload the Bean later with restored state.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Customization Overview

- Developers can modify Bean properties visually.
- Customizers or property editors enhance user experience.
- Support tailored behavior for Beans.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Creating Custom Property Editors

- Extend PropertyEditorSupport class.
- Provide GUI controls for property editing.
- Register editors using BeanInfo.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Creating Customizers

- Customizers provide a GUI to configure Beans.
- Enable more complex customization than editors.
- Useful in development environments.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Steps to Create a Simple Bean

- 1. Create a public class implementing Serializable.
- 2. Add private fields and accessor methods.
- 3. Optionally define events and BeanInfo.
- 4. Package into a JAR for reuse.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Example: TemperatureConverter Bean

- Demonstrates property handling and customization.
- Includes input/output properties and event listeners.
- Used to convert temperatures between units.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Testing Beans in BeanBox

- BeanBox is a visual tool in BDK for testing Beans.
- Allows drag-and-drop assembly of Beans.
- Supports property editing and event linking.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Connecting Beans in BeanBox

- Beans communicate via event listener registration.
- Users can connect components visually.
- No coding required for interaction.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Packaging Beans into JAR Files

- Beans are distributed as .jar archives.
- Include class files, manifests, and resources.
- Facilitates reuse and deployment.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Manifest File in JAR

- Manifest defines Bean metadata.
- Specifies main class, version, and description.
- Used by builder tools to identify Beans.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Advanced Bean Concepts

- Custom event types for specialized behavior.
- BeanContext for advanced component grouping.
- Support for pluggable look-and-feel.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Bean Persistence in Databases

- Beans can store persistent state in databases.
- Combines serialization with JDBC for data management.
- Supports enterprise-level applications.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Security in JavaBeans

- Java security model applies to Beans.
- ClassLoader restrictions and sandboxing apply.
- Ensures safe execution in visual tools.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Using Beans in Visual IDEs

- Beans integrate into tools like NetBeans or Eclipse.
- Support drag-and-drop UI assembly.
- Simplifies GUI and application design.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Benefits of Using JavaBeans

- Reusability, modularity, and ease of development.
- Platform-independent and tool-friendly.
- Simplifies complex GUI development.

Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.

Summary

- In today's lecture we have discussed about;
  - JavaBeans support reusable, component-based Java programming.
  - Core features: properties, events, introspection, persistence.
  - Enable visual assembly of applications using builder tools.

References

- Deitel, H. M., & Deitel, P. J. (2002). Advanced Java 2 Platform: How to Program (2nd ed.). Prentice Hall.