

Java Design Patterns

Este curso de Java Patterns revisa los patrones comunes y emergentes específicos del desarrollo de Java SDK y EE. Aprenderá la profundidad y la evolución de las técnicas basadas en patrones en Java, con especial énfasis en las convenciones de Java EE 6.

Aprender a:

- Distinguir entre características basadas en patrones Java EE 5 y Java EE 6.
- Implemente patrones relevantes en cada nivel del entorno Java EE.
- Vuelva a factorizar el código para mejorar las comunicaciones entre niveles.
- Relacionar el desarrollo basado en patrones con una arquitectura de implementación.
- Aplicar principios orientados a objetos y pautas de diseño.
- Implemente patrones conocidos para problemas de código específicos de Java.

Ejercicios de laboratorio

Los ejercicios de laboratorio le muestran cómo identificar, aplicar y volver a factorizar patrones seleccionados en el código, utilizando un IDE de NetBeans o Eclipse y el servidor de aplicaciones GlassFish v3. También aprenderá un subconjunto de la notación UML para acelerar la comunicación a través del diseño en lugar del código.

Patrones de diseño de Java

En los patrones de diseño, la responsabilidad de cada componente se identifica por función. Las convenciones de la documentación de patrones de diseño facilitan que los equipos de desarrollo comuniquen sus intenciones de programación y proporcionan un punto de referencia para toda la comunidad de desarrollo de Java.

Frameworks basados en Java

El lenguaje Java y los marcos populares basados en Java incorporan prácticas de desarrollo más probadas en sus interfaces de programación con cada versión importante. Estas prácticas, denominadas patrones de diseño, documentan nombres conocidos, implementación de código y técnicas de refactorización, y los riesgos y compensaciones asociados con su uso.

Prerrequisitos

Requisito previo requerido

- Desarrollo de aplicaciones para la plataforma Java EE 6 Ed 2
- Experiencia con el desarrollo de Java SE y Java EE

Java Design Patterns

Audiencia

- Desarrollador
- Integrador de sistemas

Objetivos

- Identificar los principios de diseño clave del desarrollo orientado a objetos.
- Aplicar técnicas de implementación específicas de Java a patrones conocidos
- Utilice patrones para completar el diseño de una aplicación Java
- Utilice patrones para completar el diseño de una aplicación de nivel web
- Utilice patrones para completar el diseño de una aplicación de nivel empresarial
- Utilice patrones para mejorar la comunicación entre niveles de Java EE
- Identificar y refactorizar anti-patrones en el código de trabajo
- Usando parte de un esquema de arquitectura de muestra, seleccione patrones de diseño para implementar el esquema

Temas

Revisión de los principios orientados a objetos en Java

- Describir cómo se aplican los conceptos de OO a Java.
- Describir cómo se aplican los principios de OO a Java.
- Enumere los objetivos de un lenguaje orientado a objetos
- Interpretar la notación del Lenguaje de modelado unificado (UML) y crear diagramas UML
- Identificar patrones de diseño seleccionados

Revisión de pandillas de cuatro patrones

- Enumere los patrones clave de comportamiento, creacionales y estructurales
- Aplicar el patrón de fachada
- Aplicar el patrón de estrategia
- Aplicar el patrón Observer
- Aplicar el patrón compuesto
- Revise los patrones Model-View-Controller (MVC)

Implementación de patrones en Java

- Utilice patrones de implementación diseñados para Java
- Enumere las fuerzas que afectan los patrones de clase, estado y comportamiento
- Describir en qué se diferencian los patrones, los modismos y la refactorización.

Java Design Patterns

Explorando cambios en la tecnología Java EE

- Describir los objetivos de diseño del modelo Java EE.
- Describir las mejoras en el modelo Java EE 6.

Implementar patrones de integración

- Describir patrones de diseño para el nivel de integración.
- Revise los cambios de integración de Java EE que aplican patrones de diseño
- Identificar casos de uso para aplicar patrones de niveles de integración

Implementación de patrones en componentes comerciales

- Describir el rol de un enterprise bean
- Describir patrones de diseño para el nivel empresarial.

Implementación de patrones de infraestructura en Java EE

- Describir el papel de los patrones de infraestructura de Java EE.
- Describir el patrón de Service Starter
- Describe el patrón Singleton
- Describe el patrón del localizador de frijoles
- Describe el patrón de Carpeta de recursos

Implementar más patrones de infraestructura

- Describe cómo funcionan los interceptores Java EE.
- Describir el patrón del extensor de inyección de dependencia
- Describe el patrón del extractor de carga útil
- Describir el patrón de Context Holder
- Describe el patrón de Thread Tracker

Explorando anti-patrones

- Describir la ley de abstracciones con fugas
- Definir antipatrones
- Describir AntiPatterns de nivel de integración
- Describir los antipatrones de nivel empresarial
- Describir los antipatrones del nivel de presentación

Seleccionar patrones para la arquitectura

- Definir los roles de arquitecto, diseñador y desarrollador.

Java Design Patterns

- Describir la relación entre los patrones de diseño y la arquitectura.
- Enumerar las pautas para aplicar patrones a una solución arquitectónica