

EDUCACIÓN 

Patrones de Diseño

Programa de Estudio

Patrones de Diseño

Analiza, modela y resuelve problemas de diseño de sistemas utilizando los patrones de diseño. Aprende cuándo y cómo utilizar cada uno de los patrones codificando casos de resolución real con Java y .NET, logrando un diseño profesional, maximizando la reutilización y minimizando el mantenimiento.

Contenido del curso

El curso comienza con una introducción al Lenguaje Unificado de Modelado UML repasando los conceptos básicos del diagrama de clases junto con las relaciones mas utilizadas como ser generalización, asociación, composición y agregación.

Se analizará qué son y cuándo utilizar los patrones de diseño, en particular se presentan los pertenecientes a GoF categorizados en Creacionales, Estructurales y de Comportamiento. Dentro de los Patrones Creacionales se estudian los patrones Builder, Prototype, Singleton, Abstract Factory y Factory Method. Dentro de los Patrones Estructurales se estudian los patrones Adapter, Bridge, Composite y Decorator. Y dentro de los Patrones de Comportamiento se estudian el Memento, Observer, State, Strategy, Template Method y Visitor. En cada uno de los patrones se explica cuándo utilizarlo, qué problema resuelve y cómo se implementa la solución a través de código de programación.

Finalmente se presenta el concepto de los Antipatrones, junto con su propósito y su utilización, y se realiza una introducción a otros patrones tales como Patrones de Base de Datos, Patrones de Arquitectura, Patrones de JEE y Patrones de AJAXA lo largo del curso se realizan diversos laboratorios de casos reales donde hay que analizar un problema determinado y decidir que patrón de diseño utilizar para resolverlo, y codificar su posterior solución, siendo un curso sumamente práctico.

Salida Laboral

Al finalizar el curso, el alumno se convertirá en un Diseñador de Sistemas Junior pudiendo desempeñar tareas de diseño de sistemas, focalizando en las estrategias necesarias para la

construcción de sistemas de mediana y gran envergadura.

Requisitos

Se requieren conocimientos sólidos en programación o haber realizado cualquiera de los siguientes cursos: Java J2SE Standard Programming Programación.NET con C# PHP Javascript Desarrollador Front-End Node.JS y Mongo DB Programación en iOS

Modalidad de cursado

Puedes tomar este curso en modalidad presencial o modalidad a distancia

En este curso aprenderás a

- Entender qué son los patrones de diseño y cuándo utilizarlos
- Identificar y trabajar con los distintos tipos de patrones
- Comprender qué problema resuelve y cómo se implementa la solución a través de código de programación, ya sea en tecnologías.
- Reconocer los anti-patrones, junto con su propósito y utilización
- Analizar un problema determinado y decidir qué patrón de diseño utilizar para resolverlo
- Codificar la solución de un problema en un lenguaje de programación.

Plan de Estudios



1. Introducción

Patrón: concepto

Patrones de diseño: concepto

Historia

¿Cuándo utilizarlos?

¿Cuándo no utilizarlos?

Donde utilizarlos

Que es gof

Beneficios

2. Tipos de patrones

Clasificación según su propósito

- Creacionales - De comportamiento - Estructurales

Clasificación según alcance

- De clase - De objeto

Tabla de clasificación

Especificación de patrones de diseño

3. UML review

¿Qué es UML?

Diagrama de clases

- Simbología general - Asociación - Generalización - Agregación - Composición - Multiplicidad - Ejemplo de diagrama de clases

4. Patrones de comportamiento (behavioural patterns)

Chain of responsibility pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Command pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución -

Diagrama UML

- Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Interpreter pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Iterator pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Mediator pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Memento pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Observer pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

State pattern

- Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Strategy pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Template method pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Visitor pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

5. Patrones de creación (creational patterns)

Abstract factory pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Builder pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionado

Factory method pattern

- Introducción y nombre - Intención - También conocido como - Motivación³ - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Prototype pattern

- Introducción y nombre - Intención - También conocido como: - Motivación - Solución 109 - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Singleton pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

6. Patrones de estructura (structural patterns)

Adapter pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Bridge pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Composite pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Decorator pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Façade pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Flyweight pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución 54 - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

Proxy pattern

- Introducción y nombre - Intención - También conocido como - Motivación - Solución - Diagrama UML - Participantes - Colaboraciones - Consecuencias - Implementación - Código de muestra - ¿Cuándo utilizarlo? - Patrones relacionados

7. Los anti-patrones

Anti-patrón

Historia

Propósito

Utilización

- Antipatrones de desarrollo de software - Antipatrones organizacionales

Otros patrones

- Introducción - Patrones de base de datos - Patrones de arquitectura - Patrones jee - Patrones de AJAX

EDUCACIÓN IT

Centro de Capacitación y Desarrollo Profesional



Lavalle 648 Piso 8, Microcentro, CABA

0810-220-8148

info@educacionit.com

EducaciónIT. Copyright 2005-2019