

Patrones de diseño, aplicación

Curso práctico de 5 días - 35h

Ref.: DES - Precio 2024: 2 070€ sin IVA

Este curso le formará en el diseño de aplicaciones y en prácticas de diseño modernas, como el desarrollo basado en pruebas y la refactorización. Numerosos casos prácticos le enseñarán a crear aplicaciones escalables y reutilizables, teniendo en cuenta los principales patrones de diseño.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Comprender los principios fundamentales del diseño orientado a objetos

Aplicar las reglas fundamentales para dividir una aplicación en paquetes

Aplicar los principios de construcción de clases en una aplicación

Puesta en práctica del desarrollo basado en pruebas

Implementación de los principales patrones de diseño

Los cursillistas realizarán los talleres en el lenguaje que elijan (Python, Java, C++, C# o VB.Net).

PROGRAMA

última actualización: 04/2024

1) Presentación del diseño

- Un recordatorio de los fundamentos de la programación orientada a objetos y el lenguaje UML.
- Ventajas de UML para el diseño.
- Los retos del diseño.
- Utilización de la herencia. Ventajas e inconvenientes.

2) Principios fundamentales del diseño orientado a objetos

- Los principios de apertura/cierre (OCP) y sustitución de Liskov (LSP).
- Concepto de polimorfismo, acoplamiento débil y cohesión fuerte.
- El impacto del diseño de objetos en los proyectos.

Trabajo práctico : Reparto de responsabilidades entre clases.

3) Principios de construcción de clases

- Gestión de dependencias con inversión de dependencias (DIP).
- Reducción de la complejidad aparente mediante la separación de interfaces (ISP).
- Reparto de responsabilidades con GRASP.

4) Principios de organización de paquetes

- El paquete: una unidad de diseño de entrega/reutilización (REP) y reutilización común (CRP).
- El desglose de paquetes. CCP.
- Organización entre paquetes.

Trabajo práctico : Creación y priorización de paquetes.

PARTICIPANTES

Diseñadores, desarrolladores, arquitectos o gestores de proyectos.

REQUISITOS PREVIOS

Conocimiento de un lenguaje orientado a objetos.

COMPETENCIAS DEL FORMADOR

Los expertos que imparten la formación son especialistas en las materias tratadas. Han sido validados por nuestros equipos pedagógicos, tanto en el plano de los conocimientos profesionales como en el de la pedagogía, para cada curso que imparten. Cuentan al menos con entre cinco y diez años de experiencia en su área y ocupan o han ocupado puestos de responsabilidad en empresas.

MODALIDADES DE EVALUACIÓN

El formador evalúa los progresos pedagógicos del participante a lo largo de toda la formación mediante preguntas de opción múltiple, escenificaciones de situaciones, trabajos prácticos, etc. El participante también completará una prueba de posicionamiento previo y posterior para validar las competencias adquiridas.

MEDIOS PEDAGÓGICOS Y TÉCNICOS

- Los medios pedagógicos y los métodos de enseñanza utilizados son principalmente: ayudas audiovisuales, documentación y soporte de cursos, ejercicios prácticos de aplicación y ejercicios corregidos para los cursillos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
- Al final de cada cursillo o seminario, ORSYS facilita a los participantes un cuestionario de evaluación del curso que analizarán luego nuestros equipos pedagógicos.
- Al final de la formación se entrega una hoja de presencia por cada media jornada de presencia, así como un certificado de fin de formación si el alumno ha asistido a la totalidad de la sesión.

MODALIDADES Y PLAZOS DE ACCESO

La inscripción debe estar finalizada 24 horas antes del inicio de la formación.

ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD

¿Tiene alguna necesidad específica de accesibilidad? Póngase en contacto con la Sra. FOSSE, interlocutora sobre discapacidad, en la siguiente dirección psh-accueil@orsys.fr para estudiar de la mejor forma posible su solicitud y su viabilidad.

5) Desarrollo basado en pruebas

- Desarrollo dirigido por pruebas (TDD) frente a ingeniería dirigida por modelos (MDE).
- Redacción de casos de prueba y conjuntos de pruebas.

Trabajo práctico : Asignación de responsabilidades a los componentes de software utilizando el enfoque TDD.

6) Principios de los patrones de diseño

- Patrones de diseño para reutilizar la experiencia.
- Alcance, ventajas y limitaciones de los patrones de diseño.
- Respuesta a problemas recurrentes.
- Los patrones fundacionales de Gamma y GoF: patrones de creación, comportamiento y estructura.
- Refactorización. ¿Por qué refactorizar?
- Modificación de la presentación del código y de los algoritmos de clase. Rediseño.

Trabajo práctico : Ejemplo de diseño, refactorización y programación con patrones GoF.

7) Arquitectura de software y patrones arquitectónicos

- De los requisitos a la arquitectura.
- Estilos arquitectónicos.
- Patrones de distribución (estilo cliente-servidor, patrón de bus de datos, pizarra, repositorio).
- Patrones de diseño de sistemas (MVC, arquitectura en capas, estilo Plug-in, Pipeline).

8) Proceso de desarrollo

- Diseñar en un proceso iterativo e incremental.
- El manifiesto ágil. XP, Scrum.

FECHAS

Contacto