

## OO225 Análisis y Diseño Orientado al Objeto con UML

### Descripción

- Este curso de formación trata sobre los procesos de desarrollo de software, las tecnologías de programación orientada a objetos y lenguaje unificado de modelado (UML). Mediante clases teóricas, grupos de discusión y otras actividades complementarias (como el análisis de entrevistas de interés) se proporciona una guía práctica y completa del análisis y el diseño OO, que abarca desde la recopilación de requisitos hasta la implantación del sistema.
- El curso ofrece un enfoque pragmático del desarrollo de software OO utilizando una metodología de uso generalizado (Unified Process), la última especificación de UML (versión 1.4) y tecnologías de programación OO como el lenguaje Java(TM). El curso empieza con una introducción a la tecnología OO y las metodologías de desarrollo de software para continuar con la identificación y el análisis de los requisitos (para lo cual se apoya en entrevistas con los responsables), la arquitectura y el diseño de los sistemas, la implementación y las pruebas. Durante las clases teóricas también se enseñan otras prácticas utilizadas para el análisis y el desarrollo OO, como el análisis CRC (utilizado para descubrir entidades de dominio) y el análisis de fiabilidad y solidez (utilizado para pasar del análisis al diseño).

### Temario

#### Objetivo

- Las personas que realicen el curso aprenderán a:
- Describir el proceso de desarrollo de software orientado a objetos, lo que incluye las metodologías y los flujos de trabajo de la programación OO.
- Identificar los requisitos del sistema mediante entrevistas con las personas apropiadas.
- Analizar los requisitos del sistema para determinar los casos de uso y el modelo del dominio dado (modelo de requisitos).
- Crear una arquitectura de sistemas (modelo de arquitectura) que sea compatible con los requisitos no funcionales y las restricciones del desarrollo.
- Crear un diseño de sistemas (modelo de solución) que sea compatible con los requisitos funcionales.

#### A quién va dirigido

Este curso es indicado para diseñadores de sistemas, ingenieros de software, y analistas y diseñadores de sistemas encargados de concebir y crear aplicaciones OO.

#### Pre-requisitos

Para obtener el máximo aprovechamiento del curso, los alumnos necesitan:

- Comprender los conceptos y la metodología de la programación orientada a objetos.
- Tener conocimientos generales de programación, preferiblemente con lenguaje Java.
- Conocer los fundamentos del proceso de desarrollo de sistemas.

### Cursos relacionados

1. SL-275: Programación Java (SL-275) (Antes)
2. SL-285: Desarrollo de Aplicaciones Avanzadas en Java (SL-285)
3. SL-425: Desarrollo de arquitecturas para aplicaciones empresariales Java (SL-425)
4. SL-500: Patrones de Programación J2EE (SL-500)

### Contenido

- Introducción al proceso de desarrollo del software
  1. Describir el proceso de desarrollo de sistemas orientados a objetos (OOSD).
  2. Describir la compatibilidad del lenguaje de modelado con el proceso OOSD.
  3. Explicar la finalidad, las actividades y los artefactos de los siguientes flujos de trabajo OOSD: recopilación de requisitos, análisis de requisitos, arquitectura, diseño, implementación, prueba e implantación.
- Análisis de la tecnología de diseño OO
  1. Describir cómo afectan los principios del diseño OO al proceso de desarrollo de software.
  2. Describir los principios fundamentales del diseño OO.
- Elección de una metodología de diseño OO
  1. Explicar las mejores prácticas en relación con las metodologías OOSD.
  2. Describir las funciones de varias metodologías comunes.
  3. Elegir una metodología que se adecue al proyecto.
- Determinación de la visión del proyecto
  1. Entrevistar a los propietarios del negocio para determinar los requisitos funcionales del sistema.
  2. Analizar los resultados de la entrevista para identificar los requisitos no funcionales (NFR), los riesgos y las limitaciones.
  3. Crear un documento de visión del proyecto a partir de los resultados de las entrevistas y de los análisis de riesgos.
- Identificación de los requisitos del sistema
  1. Planificar el proceso a seguir para recopilar los requisitos.
  2. Planificar entrevistas de interés para confirmar y depurar los FR y NFR en función del documento elaborado.
  3. Documentar el sistema en la especificación de requisitos del sistema (SRS) de cada uno de estos requisitos./li>
- Creación del diagrama de caso de uso inicial
  1. Identificar y describir los elementos esenciales de un diagrama de caso de uso UML.
  2. Crear un diagrama de caso de uso para el sistema a partir de SRS.
  3. Tomar nota de los escenarios de caso de uso con aspectos arquitectónicos significativos.
- Depuración del diagrama de caso de uso
  1. Documentar un caso y sus escenarios en un formulario de caso de uso.
  2. Identificar y documentar los antecedentes del caso de uso y del ejecutor.
  3. Reconocer y documentar las dependencias del caso de uso.
  4. Identificar los elementos esenciales en un diagrama de actividad.
  5. Validar un caso de uso con un diagrama de actividad.
- Determinación de las abstracciones clave
  1. Identificar una serie de posibles abstracciones clave.
  2. Identificar las abstracciones clave mediante el análisis CRC.
- Construcción del modelo de dominio dado
  1. Identificar los elementos fundamentales de un diagrama de clases UML.
  2. Construir un modelo de dominio utilizando un diagrama de clases.
  3. Identificar los elementos fundamentales en un diagrama de objetos UML.
  4. Validar el modelo de dominio con uno o varios diagramas de objetos.
- Creación del modelo de análisis con el análisis de solidez
  1. Explicar la finalidad y los elementos del modelo de diseño.
  2. Identificar los elementos fundamentales de un diagrama de colaboración UML.
  3. Crear un modelo de diseño para un caso de uso utilizando el análisis de solidez.
  4. Identificar los elementos fundamentales de un diagrama de secuencia UML.

- 5. Generar una vista de diagrama de secuencia del modelo de diseño.
- Introducción a los conceptos fundamentales de la arquitectura
  - 1. Explicar la función de la arquitectura.
  - 2. Distinguir entre arquitectura y diseño.
  - 3. Describir SunTone Architecture Methodology.
- Analizar el flujo de trabajo de la arquitectura.
  - 1. Describir el flujo de trabajo de la arquitectura.
  - 2. Describir los diagramas de las vistas de arquitectura clave.
  - 3. Seleccionar el tipo de arquitectura.
  - 4. Crear artefactos del flujo de trabajo de arquitectura.
- Creación del modelo de arquitectura para las capas cliente y presentación
  - 1. Analizar las interfaces de usuario.
  - 2. Documentar una aplicación de interfaz gráfica (GUI) en la capa cliente del modelo de arquitectura.
  - 3. Documentar una aplicación de interfaz gráfica (GUI) en la capa presentación del modelo de arquitectura.
- Creación del modelo de arquitectura para la capa negocio
  - 1. Analizar la programación orientada a objetos distribuida.
  - 2. Documentar la capa negocio del modelo de arquitectura.
- Creación del modelo de arquitectura para las capas recurso e integración
  - 1. Documentar el mecanismo de persistencia en la capa recurso del modelo de arquitectura.
  - 2. Documentar el mecanismo de persistencia en la capa integración del modelo de arquitectura.
- Creación del modelo de solución
  - 1. Crear un modelo de solución para una aplicación de interfaz gráfica.
  - 2. Crear un modelo de solución para una aplicación de interfaz de usuario web.
- Depuración del modelo de dominio
  - 1. Mejorar los atributos del modelo de dominio.
  - 2. Mejorar las relaciones del modelo de dominio.

- 3. Mejorar los métodos del modelo de dominio.
- 4. Declarar los constructores del modelo de dominio.
- Aplicación de patrones de diseño al modelo de solución
  - 1. Definir los elementos esenciales de un patrón de software.
  - 2. Describir el patrón Composite.
  - 3. Describir el patrón Strategy.
  - 4. Describir el patrón Observer.
  - 5. Describir el patrón Abstract Factory.
- Creación del modelo de estados de objeto compuestos con diagramas de estados
  - 1. Crear el modelo de estados de objeto.
  - 2. Describir las técnicas de programación de estados de objeto compuestos.

#### **Metodología**

*Curso práctico de enfoque participativo y dinámico con un estándar profesional adecuado para la formación laboral.*

#### **Requisitos de Aprobación:**

*Para aprobar el curso y recibir el Diploma, es necesario que cada participante cumpla con los siguientes requisitos: Asistencia de 75%.*

*Nota final igual o superior a 65% (escala 1 a 100) equivalente al dominio del 65% de los contenidos*

*Duración: 40 Horas cronológicas*

*Lugar: Huérfanos 835 Piso 9*

*Valor: Consulte con su Ejecutivo de Negocios, El valor del curso incluye: Material de estudio en Inglés desarrollado por Microsoft Corp. el uso de un PC por alumno, Certificado, Diploma, refrigerios.*

*Este curso cuenta con garantía de Satisfacción de Aprendizaje Golden Training se reserva el derecho de modificar la fecha de inicio de sus cursos cuando no se logre el Quórum suficiente para realizarlo*