Introducción a la programación Object/Java

Curso práctico de 3 días - 21h
Ref.: IPJ - Precio 2025: 1 300€ sin IVA

Gracias a las nociones de clase y excepción, la programación por objetos facilita la reutilización y el mantenimiento del código. Además, el uso adecuado de las interfaces permite armonizar al propietario del proyecto y al gestor del mismo. Utilizando varios programas Java, este curso detalla todos los principios fundamentales del desarrollo de objetos.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Comprender los principios fundadores de Objet

Aprender la sintaxis del lenguaje Java

Dominar los intercambios técnicos con los equipos de desarrollo

Dominio de la construcción de especificaciones funcionales de tipo objeto

PROGRAMA

última actualización: 05/2024

1) Presentación general

- Principios fundadores del Objeto: abstracción/encapsulación. Herencia, implementación.
- Presentación general: el lenguaje, las herramientas, la biblioteca.
- Distribuciones Java.

2) Sintaxis, tipos y expresiones

- Estructuración sintáctica de una aplicación Java.
- Ejemplo de sintaxis en una aplicación simplificada.
- Vista externa de una clase: sintaxis de uso.
- Vista interna de una clase: sintaxis de implementación.
- Noción de tipo. Comparación del uso de tipos básicos y tipos objeto.
- Uso sencillo de los tipos básicos: enteros, flotantes, Char y Boolean.
- Noción de expresión.
- Ejemplos de declaraciones: variables y constantes.
- Denominación comparativa de tipos básicos y tipos de objeto.
- Utilización de operadores con objetos.
- Campos estáticos o variables de clase.
- Información adicional sobre tipos: uso básico de matrices.
- Conversión tipo básico/tipo objeto.
- Convenciones de escritura.

3) Métodos e instrucciones

- Sintaxis de invocación de métodos.
- Métodos de clase y métodos de instancia.
- Definición y utilización de métodos.
- Sobrecarga de métodos.
- Noción de subbloque.
- Categorías de instrucción.
- Principales instrucciones de control: if, while, for, return, break.

PARTICIPANTES

Jefes de proyecto que deseen conocer con precisión las técnicas utilizadas por sus colegas o subcontratistas. Desarrolladores tradicionales que deseen pasar al desarrollo orientado a objetos.

REQUISITOS PREVIOS

Conocimientos básicos de programación.

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.

4) Utilizar la abstracción

- Un ejemplo sencillo de utilización de un objeto: declaración, instanciación o creación, delegación.
- Uso de constructores de objetos: descubre la documentación en línea.
- Utilización de la interfaz programática de objetos: ejemplo de la clase Date.
- Una clase muy utilizada: la clase String.
- Particularidades relativas a las cadenas de caracteres.
- Utilización de la clase StringBuffer: un ejemplo de uso de la sobrecarga de métodos.

5) Utilizar la herencia

- Recordatorio del principio de herencia y terminología.
- Utilizar la herencia.
- Ejemplo de gráfico de herencia.
- La clase Objeto y la genericidad.
- Uso del polimorfismo.
- Especialización de una referencia polimórfica.
- Tipificación de referencias/tipificación de objetos.
- Comportamiento del método y mecanografía.
- Genericidad de las clases contenedoras: ejemplo de la clase Vector.
- Las adiciones de JAVA 5 (TIGER): genéricos.

6) Utilizar el mecanismo de interfaz

- Interfaz implícita y explícita de una clase.
- Sintaxis asociada a interfaces explícitas.
- Casos de uso de las referencias a interfaces: flexibilidad, ámbito limitado, polimorfismo.
- Ejemplo de implementación de múltiples interfaces.
- Síntesis sobre el interés de las interfaces para los métodos.
- Uso de interfaces para constantes.
- Ejemplos avanzados de uso de interfaces.

7) Desarrollo de clases

- Enfoque metodológico, análisis estático, dinámico y empresarial.
- Notación UML: diagramas de clase, estado/transición y secuencia.
- Esqueleto de clase: componentes básicos, herramientas de generación automática.
- Información adicional sobre derechos de acceso.
- Organizados en paquetes.
- Limitaciones del paquete.
- Constructores de escritura.
- Constructor por defecto.
- Información adicional sobre la escritura de constructores.
- La autorreferencia "esto".
- Campos y métodos estáticos.
- El método principal.

8) Desarrollo de interfaces

- Recordatorios y adiciones a los principios.
- Sintaxis asociada a las interfaces, caso de las constantes.
- Definición de interfaces para métodos.
- Múltiples implementaciones de interfaces y extensiones.
- Implementación parcial de la interfaz.
- Ejemplos de uso de interfaces.

9) Desarrollo de clases derivadas

- Un recordatorio de los principios.
- Enfoque metodológico de la división en clases.

- Métodos y clases abstractas.
- Clases abstractas e interfaces.
- Derecho de acceso a los campos y a la herencia.
- Vinculación de constructores y herencia.
- Redefinición y sobrecarga.

10) Excepciones

- Principios y cinemática general.
- Detección, registro y notificación de una situación excepcional.
- Aplazamiento de una excepción: cláusula Throws en la firma, bloque Try/Catch.
- Excepciones no verificadas.
- Ejemplo con gestión de excepciones.

FECHAS

Contacto