

Conceptos básicos de Java y objetos

Curso práctico de 5 días - 35h

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

Este curso introduce a los estudiantes en el lenguaje Java y la programación de objetos presentando gradualmente todas las construcciones del lenguaje. El curso también abarca los problemas de diseño (utilizando la notación UML) y las principales bibliotecas estándar: entrada/salida, colecciones e interfaces gráficas.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Aplicar los principios de la programación orientada a objetos

Dominar la sintaxis del lenguaje Java

Dominar las principales bibliotecas estándar de Java

Dominio de un entorno de desarrollo integrado para programar en Java

Los ejercicios prácticos se han diseñado para ilustrar todos los elementos del lenguaje y poner en práctica los conceptos del diseño orientado a objetos: todos los ejercicios incluyen una fase de análisis/diseño seguida de una fase de programación.

PROGRAMA

última actualización: 09/2023

1) Técnicas de objetos

- Los principios generales del modelado y la programación de objetos.
- Abstracción y encapsulación: interfaces.
- Las diferentes formas de herencia, polimorfismo.
- Introducción a la modelización UML: el modelo estático, el modelo dinámico, el modelo de cooperación, los escenarios.

Trabajo práctico : La especificación UML de un caso práctico, que será uno de los principios rectores de los ejercicios siguientes.

2) Construcciones básicas del lenguaje

- Variables: declaración y tipificación.
- Definición de campos.
- Métodos: definición.
- Expresiones.
- Instrucciones de control: instrucciones condicionales, de bucle y de bifurcación.
- Pinturas.
- Tipos enumerados, autoboxing.
- Unidades de compilación y paquetes: control de la visibilidad de las clases, mecanismo de importación.
- Importaciones estáticas.

Trabajo práctico : Una serie de ejercicios sencillos para familiarizarse con el entorno de desarrollo y crear un programa sencillo. Utilización de paquetes.

3) Definición e instanciación de clases

- Clases y objetos.
- Campos y métodos.
- Fabricantes.
- Autorreferencia.

PARTICIPANTES

Desarrolladores sin experiencia, ingenieros, gestores de proyectos cercanos al desarrollo.

REQUISITOS PREVIOS

Conocimientos básicos de programación. Deseable experiencia en desarrollo de aplicaciones.

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.

- Campos y métodos estáticos.
- Métodos con un número variable de argumentos.
- Aspectos metodológicos: diseño de la clase.

Trabajo práctico : Programación del estudio de casos.

4) El legado

- Las distintas formas de herencia: extensión e implementación.
- Interfaces e implementación de interfaces. Polimorfismo y su implementación.
- Extensión. Definición de clases derivadas, constructores y referencias. Aspectos metodológicos.
- Construcción de jerarquías de clases. Factorización del código: clases abstractas.
- Uso simultáneo de implementación y extensión. Clases abstractas.
- Aspectos metodológicos: agrupación de constantes, especificación de servicios.
- Construcción de jerarquías de clases e interfaces.
- Genericidad.

Trabajo práctico : Diseñar y construir una jerarquía de clases e interfaces. Implementar el polimorfismo y la genericidad en el caso práctico.

5) Excepciones

- Bloques de prueba, generación de excepciones.
- El algoritmo de selección catch().
- Aspectos metodológicos: construcción de una jerarquía de excepciones, utilización de excepciones.

Trabajo práctico : Introducción de excepciones en el estudio de casos.

6) Programación de entradas y salidas

- La jerarquía de las clases de entrada/salida.
- Algunas clases para manipular sistemas de archivos.
- Algunas clases de entrada/salida que trabajan con flujos byte y flujos Char.
- Entrada/salida del teclado.

Trabajo práctico : Leer/escribir archivos.

7) Programación gráfica

- Conceptos básicos: los principios de visualización y gestión de eventos desde jdk1.1.
- Visualización de componentes gráficos: contenedores y diseños.
- Algunos componentes gráficos: etiquetas, botones, zonas de texto.
- Gestión de eventos: escuchadores y adaptadores. Asociación de manejadores con componentes gráficos.

Trabajo práctico : Construir una interfaz gráfica de usuario.

8) Algunas clases de utilidad

- Clases de sistemas.
- Clases de contenedores.

Trabajo práctico : Implementación de clases de utilidad.

FECHAS

Contacto