

Introducción a la programación (con Java)

Curso práctico de 3 días - 21h

Ref.: INJ - Precio 2025: 1 280€ sin IVA

Este curso te introduce en la programación tradicional (desde las variables hasta el acceso al SGBD) y te ofrece una breve introducción a la programación de objetos. Muchos de los programas que crearás son en Java, pero los conceptos y métodos que aprenderás seguirán siendo válidos independientemente de los lenguajes que utilices en tus futuras tareas.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Estructurar programas según un algoritmo

Dominar el léxico y la sintaxis de un lenguaje para escribir un programa

Compilar y ejecutar un programa

Depurar y probar un programa

Acceder a una base de datos

Comprender los principios fundamentales de la programación orientada a objetos

Este curso contiene más de un 60 % de ejercicios prácticos realizados en Visual Basic (ref. curso INP), Java (ref. curso INJ), C# (ref. curso OGR) o Python (ref. curso THO).

PROGRAMA

última actualización: 04/2024

1) Los fundamentos de la programación

- ¿Qué es un programa? ¿Qué es un lenguaje? Los distintos paradigmas. ¿Qué lenguaje para qué aplicación?
- Los compiladores. Los ejecutables.
- Las responsabilidades de un programador.
- ¿Qué es un algoritmo?
- Las necesidades a las que responde un algoritmo.
- El concepto de pseudolenguaje.

Trabajo práctico : Presentación de diferentes lenguajes (Java, C#, Visual Basic, C, C++).

Escritura de un algoritmo inicial en pseudolenguaje.

2) Generación de un primer programa

- Escritura de un programa sencillo: sintaxis e instrucciones.
- Compilación y ejecución del programa.
- ¿Qué es una biblioteca? Su función y su uso.

Trabajo práctico : Descubrir el entorno de desarrollo y ejecución. Escribir, compilar y ejecutar un primer programa.

3) Reglas de programación

- Convención de nombres.
- Convenciones sintácticas.
- Utilización de comentarios. ¿Por qué comentar los desarrollos?
- Mejorar la legibilidad de los programas: sangrar el código, dividir el código, etc.

PARTICIPANTES

Cualquier persona que necesite aprender a programar.

REQUISITOS PREVIOS

No se requieren conocimientos especiales.

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) Variables

- ¿Qué es una variable?
- ¿Por qué digitar una variable?
- Tipos primitivos: enteros, cadenas de caracteres, números reales, otros.
- Declaración, definición e inicialización de una variable.
- Constantes.
- Introducción, visualización, asignación, conversión de tipos.
- Organizar datos en matrices.
- Tipos avanzados: registro, matriz, árbol.

Trabajo práctico : Escribir varios programas sencillos que manipulen variables.

5) Operadores y expresiones

- Los diferentes operadores (multiplicativo, aditivo, comparación, igualdad, lógico, asignación).
- Combinación de operadores.
- Expresiones booleanas.

Trabajo práctico : Manejo de operadores y expresiones booleanas.

6) Estructuras de control

- Selecciones alternativas (si, si-entonces-sino, selección de caso).
- Bloques de instrucciones (concepto de Inicio... Fin).
- Bucles iterativos (siempre que-repetir, repetir-hasta, para-de- a).
- Anidamiento de instrucciones.
- Comentarios.

Trabajo práctico : Utilización de estructuras de control para implementar un algoritmo.

7) Procedimientos y funciones

- Definiciones: procedimiento, función.
- ¿Por qué son esenciales en programación (reutilización, legibilidad, etc.)? ?
- Paso de parámetros.
- El código de retorno de una función.
- Conocimiento de los límites del paso del valor de una variable.
- Concepto de paso por dirección.
- Llamada a funciones.

8) Introducción a la programación de Objetos

- Conceptos asociados a la programación de Objetos: clase, atributo, método, argumento.
- Modelización de Objetos basada en requisitos funcionales.
- Introducción a las buenas prácticas en el diseño y organización de un programa.

Trabajo práctico : Ilustración de los conceptos de Objetos.

9) Acceso a bases de datos

- Organización y almacenamiento de datos.
- Tratamiento básico (conexión, consultas, recuperación de datos).
- Aplicación cliente y servidor de datos.
- Visualización y manipulación de los datos en la aplicación cliente.

Trabajo práctico : Creación de un formulario de búsqueda de información en una base de datos.

10) Mantenimiento, depuración y comprobación de programas

- Saber leer e interpretar los diferentes mensajes de error.
- Utilizar un depurador: ejecución de un programa paso a paso, puntos de interrupción, inspección de variables durante la ejecución.
- Planificación de pruebas unitarias.

Trabajo práctico : Utilización de un depurador para controlar la ejecución de un programa.

FECHAS

Contacto