



Principios de la programación orientada a objetos

Modalidad:

e-learning con una duración 56 horas

Objetivos:

Dominar los conceptos fundamentales del paradigma orientado a objetos
Desarrollar clases aplicando los fundamentos del paradigma Orientado a Objetos

Contenidos:

Tema 1. Introducción al paradigma orientado a objetos.

- 1.1 Ciclo de desarrollo del software bajo el paradigma de orientación a objetos: Análisis, diseño y programación orientada a objetos.
- 1.2 Análisis del proceso de construcción de software: Modularidad.
- 1.3 Distinción del concepto de módulo en el paradigma orientado a objetos.
- 1.4 Identificación de objetos como abstracciones de las entidades del mundo real que se quiere modelar.

Tema 2. Clases y objetos.

- 2.1 Distinguir el concepto de clase y sus atributos, métodos y mecanismo de encapsulación.
- 2.2 Análisis de los objetos: Estado, comportamiento e identidad.
- 2.3 Uso de objetos como instancias de clase. Instancia actual (This, Self, Current).
- 2.4 Identificación del concepto de programa en el paradigma orientado a objetos. POO = Objetos + Mensajes.

Tema 3. Generalización/Especialización: herencia.

- 3.1 Descripción del concepto de herencia: Simple y múltiple.
- 3.2 Distinción de la herencia múltiple.
- 3.3 Creación de objetos en la herencia.
- 3.4 Clasificación jerárquica de las clases.





Tema 4. Relaciones entre clases.

- 4.1 Distinción entre Agregación/Composición.
- 4.2 Distinción entre Generalización / Especialización.
- 4.3 Identificación de asociaciones.

Tema 5. Análisis del polimorfismo.

- 5.1 Concepto.
- 5.2 Tipos.
- 5.3 Polimorfismo en tiempo de compilación (Sobrecarga).
- 5.4 Polimorfismo en tiempo de ejecución (Ligadura Dinámica).
- 5.6 Objetos polimórficos.
- 5.7 Comprobación estática y dinámica de tipos.

Tema 6. Técnicas de programación estructurada.

- 6.1 Identificación de elementos básicos: constantes, variables, operadores y expresiones.
- 6.2 Análisis de estructuras de control: Secuencial, condicional y de repetición.
- 6.3 Distinción entre funciones y procedimientos.
- 6.4 Demostración de llamadas a funciones y procedimientos.
- 6.5 Empleo de llamadas a funciones y procedimientos incluidos en las clases.

Tema 7. Estructura de la información.

- 7.1 Enumeración de datos simples: Numéricos (enteros y reales), lógicos, carácter, cadena de caracteres, puntero o referencia a memoria.
- 7.2 Datos estructurados: Arrays.
- 7.3 Mecanismos de gestión de memoria.

Tema 8. Lenguajes de programación orientados a objetos.

- 8.1 Análisis del lenguaje de programación orientado a objetos y paradigma orientado a objetos.
- 8.2 Comparación entre los lenguajes de programación orientados a objetos más habituales.
Características esenciales.
- 8.3 Librerías de clases.

Tema 9. Implementación del paradigma utilizando un lenguaje de programación orientado a objetos.

- 9.1 Elección del lenguaje.
- 9.2 Enumeración de los tipos de aplicaciones.
- 9.3 Herramientas de desarrollo.
- 9.4 Tipos de datos y elementos básicos característicos del lenguaje. Instrucciones.
- 9.5 Estudio y utilización de las clases básicas incluidas en la librería de clases.





- 9.6 Definición de clases.
- 9.7 Construcción de métodos. Sobrecarga.
- 9.8 Construcción de atributos.
- 9.9 Construcción de la interfaz de la clase.
- 9.10 Construcción de clases incluyendo relaciones de Agregación /Composición y Asociación.
- 9.11 Construcción de clases con herencia.
- 9.12 Construcción de clases con herencia múltiple.
- 9.13 Definición de clases abstractas.
- 9.14 Construcción de clases con herencia incluyendo poliformismo.
- 9.15 Empleo de excepciones.
- 9.16 Gestión de eventos.
- 9.17 Empleo de hilos.
- 9.18 Definición y análisis de programación en red.
- 9.19 Acceso a bases de datos desde las aplicaciones. Librerías de clases asociadas.

