

# Principios de la programación orientada a objetos

## Modalidad:

e-learning con una duración 56 horas

## Objetivos:

Permitirá al alumnado adquirir conocimientos sobre los principios de la programación orientada a objetos.

## Contenidos:

### UNIDAD DIDÁCTICA 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.

1.4.1 Descripción de objetos: Conjunto de datos que definen un objeto y conjunto comportamientos que pueden solicitarse a los objetos.

1.4.2 Identificación del comportamiento de un objeto: Concepto del mensaje.

### UNIDAD DIDÁCTICA 2. Clases y objetos

2.1 Distinguir el concepto de clase y sus atributos, métodos y mecanismo de encapsulación:



La manera más sencilla de que crezca  
tu organización

**CON LOS LÍDERES EN  
FORMACIÓN**



2.1.1 Relación entre interfaz y clase

2.1.2 Distinción de los tipos de datos y clases

2.2 Análisis de los objetos: Estado, comportamiento e identidad:

2.2.1 Análisis de mensajes

2.2.2 Tipos de métodos y su clasificación: Métodos de acceso, de selección o consulta, de construcción, de destrucción.

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.

**UNIDAD DIDÁCTICA 3. Generalización/Especialización: herencia**

3.1 Descripción del concepto de herencia: Simple y múltiple:

3.1.1 Relación de herencia: Características.

3.1.2 Reglas y características que definen una relación de herencia: Regla ¿Es- un?

3.1.3 Transmisión de atributos y métodos.

3.1.4 Regla de especialización de la superclase en la subclase

3.1.5 Acceso a los atributos de una clase y acoplamiento entre las clases

3.1.6 Utilización de objetos this (current, self u otros) y super.

3.1.7 Leyes de Demeter

3.2 Distinción de la herencia múltiple:

3.2.1 Problemas: Conflictos de nombres, herencia repetida.

3.2.2 Soluciones



La manera más sencilla de que crezca  
tu organización

**CON LOS LÍDERES EN  
FORMACIÓN**



3.3 Creación de objetos en la herencia

3.4 Clasificación jerárquica de las clases:

3.4.1 Clase raíz

3.4.2 Clases abstractas

3.4.3 Métodos virtuales

3.4.4 Redefinición de métodos

UNIDAD DIDÁCTICA 4. Relaciones entre clases

4.1 Distinción entre Agregación/Composición.

4.2 Distinción entre Generalización / Especialización.

4.3 Identificación de asociaciones

UNIDAD DIDÁCTICA 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

UNIDAD DIDÁCTICA 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.



La manera más sencilla de que crezca  
tu organización

**CON LOS LÍDERES EN  
FORMACIÓN**



## 6.3 Distinción entre funciones y procedimientos:

### 6.3.1 Interfaz

### 6.3.2 Paso de parámetros: por valor y por referencia

### 6.3.3 Parámetros actuales y formales

### 6.3.4 Funciones: valor de retorno

### 6.3.5 Procedimientos

### 6.3.6 Ámbito de las variables

### 6.3.7 Almacenamiento de las variables

## 6.4 Demostración de llamadas a funciones y procedimientos.

## 6.5 Empleo de llamadas a funciones y procedimientos incluidos en las clases:

### 6.5.1 Llamadas calificadas y no calificadas (instancia actual)

### 6.5.2 Paso de parámetros

### 6.5.3 los atributos de la clase

## UNIDAD DIDÁCTICA 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:

#### 7.2.1 Listas enlazadas, pilas y colas

#### 7.2.2 Estructuras

#### 7.2.3 Ficheros



#### 7.2.4 Otras estructuras complejas: Tablas hash e introducción a los árboles y grafos

#### 7.3 Mecanismos de gestión de memoria:

##### 7.3.1 Uso de la gestión automática de memoria

##### 7.3.2 Construcción y destrucción de objetos

##### 7.3.3 Objetos inalcanzables

##### 7.3.4 Recolección de «basura».

##### 7.3.5 Métodos constructores y destructores.

#### UNIDAD DIDÁCTICA 8. Lenguajes de programación orientados a objetos

#### 8.1 Análisis del lenguaje de programación orientado a objetos y paradigma orientado a objetos:

##### 8.1.1 Lenguajes de programación orientados a objetos

##### 8.1.2 Lenguajes de programación basados en objetos

##### 8.1.3 Lenguajes de programación que utilizan objetos

#### 8.2 Comparación entre los lenguajes de programación orientados a objetos más habituales. Características esenciales.

#### 8.3 Librerías de clases:

##### 8.3.1 Definición de su estructura

##### 8.3.2 Creación y utilización

#### UNIDAD DIDÁCTICA 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.

La manera más sencilla de que crezca  
tu organización

**CON LOS LÍDERES EN  
FORMACIÓN**



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:

9.16.1 Eventos, fuentes y auditores de eventos

9.16.2 Tipos de eventos. Mecanismos de gestión de eventos

9.16.3 Librerías de clases asociadas

9.17 Empleo de hilos:

9.17.1 Fundamentos

9.17.2 Creación



La manera más sencilla de que crezca  
tu organización

**CON LOS LÍDERES EN  
FORMACIÓN**



9.17.3 Prioridad

9.17.4 Comunicación

9.17.5 Sincronización

9.17.6 Estados

9.17.7 Creación y ejecución de hilos en el lenguaje

9.17.8 Librerías de clases asociadas

9.17.9 Programación multihilo

9.18 Definición y análisis de programación en red:

9.18.1 Aplicación cliente servidor

9.18.2 Sockets

9.19 Acceso a bases de datos desde las aplicaciones. Librerías de clases asociadas