

ztudia<sup>\*</sup>

PROGRAMACIÓN ORIENTADA A OBJETOS

\* Ztudía es una solución tecnológica propiedad de Five Digital Minds. Su registro de marca está en proceso.



## CERTIFICACIÓN ZTUDIA EN PROGRAMACIÓN ORIENTADA A OBJETOS.

### OBJETIVO DEL CURSO

Al finalizar el curso, el Ztudiante será capaz de aplicar un conocimiento amplio de los principios fundamentales de la Programación Orientada para crear y utilizar clases y objetos básicos, lo que le permitirá modelar problemas sencillos del mundo real utilizando el paradigma de objetos y comprender los conceptos globales del manejo de excepciones y colecciones.

2

### 1. Introducción a la Programación Orientada a Objetos

- 1.1. Conocer la historia de la programación
- 1.2. Analizar el concepto de programa
- 1.3. Analizar el concepto de lenguaje de programación
- 1.4. Entender la programación estructurada
- 1.5. Conocer el paradigma de Programación Orientada a Objetos
- 1.6. Diferenciar entre programación estructurada y POO
- 1.7. Conocer los entornos de desarrollo integrado
- 1.8. Dividir problemas complejos en subproblemas

### 2. El principio de Abstracción

- 2.1. Identificar la base de un conjunto
- 2.2. Comprender el concepto de Abstracción
- 2.3. Identificar las propiedades de una abstracción
- 2.4. Identificar las acciones de una abstracción
- 2.5. Abstractar la realidad a sus formas básicas
- 2.6. Abstractar conceptos a sus formas básicas

### 3. Lenguaje de Modelado Unificado

- 3.1. Comprender el concepto de Lenguaje de Modelado Unificado
- 3.2. Identificar los elementos UML útiles en la POO
- 3.3. Analizar la composición de una clase UML
- 3.4. Crear una clase UML
- 3.5. Identificar superclases en UML
- 3.6. Identificar subclases en UML
- 3.7. Modelar relaciones entre clases UML
- 3.8. Analizar el Diagrama de Clases UML

### 4. Creación de clases y objetos

- 4.1. Comprender el concepto de Clase

- 4.2. Analizar los atributos de una clase
- 4.3. Analizar los métodos de una clase
- 4.4. Analizar el método constructor de una clase
- 4.5. Modificar el acceso a una clase
- 4.6. Comprender el concepto de Objeto
- 4.7. Crear objetos

## 5. Uso de clases y objetos

- 5.1. Acceder a los atributos de un objeto
- 5.2. Acceder a los métodos de un objeto
- 5.3. Utilizar objetos para resolver necesidades específicas
- 5.4. Sobrecargar el método constructor de una clase
- 5.5. Crear objetos mediante constructores sobrecargados
- 5.6. Utilizar la palabra especial THIS / SELF
- 5.7. Utilizar la palabra especial STATIC
- 5.8. Destruir objetos

## 6. El principio de Encapsulamiento

- 6.1. Comprender el concepto de Encapsulamiento
- 6.2. Encapsular atributos
- 6.3. Encapsular métodos
- 6.4. Comprender el método especial getter
- 6.5. Comprender el método especial setter
- 6.6. Utilizar el método especial getter
- 6.7. Utilizar el método especial setter

## 7. El principio de Herencia

- 7.1. Comprender el concepto de Clase Base
- 7.2. Comprender el concepto de Clase Derivada
- 7.3. Comprender el concepto de Herencia
- 7.4. Crear clases con herencia
- 7.5. Crear clases con herencia múltiple
- 7.6. Acceder a los atributos de una clase base
- 7.7. Acceder a los métodos de una clase base

## 8. El principio de Polimorfismo

- 8.1. Comprender el concepto de Polimorfismo
- 8.2. Sobrecargar métodos

- 8.3. Sobrescribir métodos
- 8.4. Comprender el concepto de interfaz
- 8.5. Crear interfaces
- 8.6. Comprender el concepto de clase abstracta
- 8.7. Crear clases abstractas

## 9. Manejo de excepciones

- 9.1. Comprender el concepto de excepciones
- 9.2. Crear bloques TRY - CATCH
- 9.3. Definir un bloque de sentencias TRY
- 9.4. Definir un bloque de sentencias CATCH
- 9.5. Identificar los diferentes tipos de excepciones
- 9.6. Definir el tipo de excepción que procesará el bloque TRY - CATCH

## 10. Creación de clases y objetos avanzados

- 10.1. Sobrecargar el método constructor con parámetros de tipo personalizado
- 10.2. Crear objetos utilizando parámetros de tipo personalizado
- 10.3. Devolver la información de un objeto asociado por medio de otro objeto
- 10.4. Crear clases internas
- 10.5. Acceder a los atributos de una clase interna
- 10.6. Acceder a los métodos de una clase interna
- 10.7. Crear clases genéricas
- 10.8. Crear objetos a partir de clases genéricas
- 10.9. Crear clases finales

## 11. Manejo de colecciones y bibliotecas

- 11.1. Crear listas
- 11.2. Utilizar listas
- 11.3. Crear diccionarios
- 11.4. Utilizar diccionarios
- 11.5. Crear conjuntos
- 11.6. Utilizar conjuntos

## 12. Introducción a los patrones de diseño

- 12.1. Comprender qué son los patrones de diseño
- 12.2. Conocer el patrón de diseño creacional SINGLETON
- 12.3. Conocer el patrón de diseño creacional FACTORY METHOD
- 12.4. Conocer el patrón de diseño estructural ADAPTER



- 12.5. Conocer el patrón de diseño estructural FACADE
- 12.6. Conocer el patrón de diseño comportamental OBSERVER
- 12.7. Conocer el patrón de diseño comportamental STRATEGY

### 13. Principios SOLID

- 13.1. Conocer el Single Responsibility Principle (SRP)
- 13.2. Conocer el Open/Closed Principle (OCP)
- 13.3. Conocer el Liskov Substitution Principle (LSP)
- 13.4. Conocer el Interface Segregation Principle (ISP)
- 13.5. Conocer el Dependency Inversion Principle (DIP)

### 14. Buenas prácticas

- 14.1. Utilizar las convenciones de codificación
- 14.2. Codificar en modo estricto
- 14.3. Analizar el principio Don't Repeat Yourself (DRY)
- 14.4. Analizar el principio Keep It Simple, Stupid (KISS)
- 14.5. Analizar el principio You Aren't Gonna Need It (YAGNI)
- 14.6. Documentar el código
- 14.7. Controlar las versiones de un programa