

# UD5: Programación Orientada a Objetos avanzada.

JOSÉ CLIMENT PENADÉS

# Índice

- ▶ Ubicación y justificación
- ▶ Normativa
- ▶ Objetivos
- ▶ Competencias
- ▶ Contenidos
- ▶ Metodología
- ▶ Agrupamientos
- ▶ Sesiones
- ▶ Evaluación
- ▶ Atención a la diversidad
- ▶ Recursos y bibliografía
- ▶ Actividades

# Ubicación y justificación

- ▶ 2º Evaluación,.
- ▶ 40 horas = 20 días (2 horas por día)
- ▶ RA 4: Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos
- ▶ RA 7: Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.
- ▶ Software propio industrial.
- ▶ Paradigma actual mas extendido

# Normativa

- ▶ Ley Orgánica de educación (2/2006)
- ▶ Ley Orgánica de Modificación de la LOE (3/2020)
- ▶ Ley Orgánica de la Formación Profesional (3/2022)
- ▶ Real decreto del calendario de implantación (278/2023)
- ▶ Instrucciones para el alumnado procedente de otros países(16/07/2019)
- ▶ Orden sobre la prevención de riesgos laborales (1/2014)
- ▶ Instrucciones de inicio de curso (08/08/2024)
- ▶ Real decreto de ordenación del Sistema de Formación Profesional.(659/2023)
- ▶ Título del CFGS DAM (450/2010)
- ▶ Actualización del CFGS DAM(405/2023)
- ▶ Modificación del CFGS DAM (500/2024)
- ▶ Currículo nacional del CFGS DAM (EDU/2000/2010)
- ▶ Currículo autonómico del CFGS DAM (58/2012)
- ▶ Decreto de equidad e inclusión(104/2018)
- ▶ Orden de respuesta educativa a la inclusión(20/2019)
- ▶ Decreto de igualdad y convivencia (195/2022)

# Objetivos

Objetivos	
OD1	Aplicar los conceptos de encapsulación y control de visibilidad para limitar el acceso a los datos.
OD2	Diseñar e implementar métodos estáticos, comprendiendo su alcance y utilidad.
OD3	Aplicar la herencia para definir jerarquías de clases que favorezcan la reutilización del código.
OD4	Implementar polimorfismo y la sobreescritura de métodos para adaptar el comportamiento según el contexto.
OD5	Utilizar composición y agregación para modelar relaciones complejas entre objetos.
OD6	Emplear interfaces para definir contratos de comportamiento que aseguren la interoperabilidad.
OD7	Organizar y modularizar el código mediante empaquetado de clases, facilitando su mantenimiento.
OD8	Documentar y evaluar diseños avanzados mediante métodos de depuración y análisis crítico.

# Competencias

- ▶ Competencias pps:
  - ▶ C1: Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías y herramientas adecuados a las especificaciones.
  - ▶ C3: Establecer vías eficaces de relación profesional y comunicación con sus superiores, compañeros y subordinados, respetando la autonomía y competencias de las distintas personas.
  - ▶ C4: Mantener el espíritu de innovación y actualización en el ámbito de su trabajo para adaptarse a los cambios tecnológicos y organizativos de su entorno profesional.

# Contenidos

4.8. Utilización de clases y objetos.  
4.9. Utilización de clases heredadas.  
4.10. Codificación y utilización de métodos estáticos.  
4.11. Concepto y utilización de Interfaces.  
4.12. Empaquetados de clases.  
Organización de las clases en paquetes.

## 7. Utilización avanzada de clases.

7.1. Composición de clases.  
7.2. Herencia.  
7.3. Jerarquía de clases:  
Superclases y subclases.  
7.4. Clases y métodos abstractos y finales.  
7.5. Sobreescritura de métodos.  
7.6. Constructores y herencia.

7.7. Acceso a métodos de la superclase.  
7.8. Polimorfismo.  
7.9 Interfaces  
7.10. Codificación, pruebas, depuración y documentación de programas que implementen jerarquías de clases.

# Metodología

- ▶ Práctica y motivadora
- ▶ Orientaciones metodológicas:
  - ▶ Conocimientos previos del alumnado
  - ▶ Aprendizaje significativo y funcional
  - ▶ Metodologías activas



# Actividades E/A

- ▶ Actividades de presentación
- ▶ Actividades de evaluación de conocimientos previos
- ▶ Actividades de desarrollo de contenidos
- ▶ Actividades de cierre o síntesis-resumen
- ▶ Actividades de diagnóstico
- ▶ Actividades de refuerzo y ampliación
- ▶ Actividades evaluables o de evaluación

# Agrupamientos

- ▶ Trabajo individual
- ▶ Trabajo cooperativo en grupos pequeños
- ▶ Trabajos grupo-clase

# Sesiones

Planificación de las sesiones	
S1	<ul style="list-style-type: none"><li>- Presentación de los nuevos conceptos y repaso de los conceptos básicos.</li><li>- Ejercicios de refuerzo.</li></ul>
S2 – S5	<ul style="list-style-type: none"><li>- Teoría sobre encapsulación, visibilidad, composición y empaquetado de clases. y control de visibilidad.</li><li>- Ejercicios sobre los conceptos trabajados.</li></ul>
S6 – S8	<ul style="list-style-type: none"><li>- Teoría sobre la herencia, diseño de jerarquías y extensión de clases. Superclase y subclase.</li><li>- Ejercicios prácticos de herencia.</li></ul>
S9 – S13	<ul style="list-style-type: none"><li>- Uso de las clases heredadas, constructores y acceso a los métodos de la superclase.</li><li>- Ejercicios avanzados de herencia.</li></ul>
S14 – S17	<ul style="list-style-type: none"><li>- Polimorfismo y sobrescritura de métodos.</li><li>- Ejercicios de polimorfismo.</li></ul>
S18 – S20	<ul style="list-style-type: none"><li>- Teoría y demostraciones sobre métodos estáticos; análisis de casos de uso.</li><li>- Ejercicios y ejemplos sobre los métodos estáticos.</li></ul>

# Sesiones

S21 S23	–	- Taller práctico para la implementación y utilización de métodos estáticos; ejercicios en el IDE.
S24 S26	–	- Explicación de las clases y métodos abstractos y finales. - Ejemplos sobre los contenidos vistos.
S27 S28	–	- Laboratorio práctico en herencia, composición, agregación y polimorfismo mediante ejercicios aplicados. - Actividades de ampliación y refuerzo sobre los conceptos vistos durante la unidad.
S29 S32	–	- Uso de interfaces: definición y aplicación de comportamientos comunes en diferentes clases. - Ejercicios con interfaces.
S33 S37	–	- Sesiones de practica con casos prácticos reales y debate crítico; retroalimentación y ajustes finales.
S38 S40	–	- Actividades de ampliación y refuerzo. - Cuestionario sobre los conceptos vistos durante la unidad. - Resolución de dudas.

# Evaluación

## ▶ Proceso de enseñanza

- ▶ Autoevaluación de la metodología
- ▶ A mitad de la unidad y al final
- ▶ Rubrica de autoevaluación y cuaderno del docente

## ▶ Proceso de aprendizaje

- ▶ Criterios de evaluación del RA
- ▶ Evaluación inicial, continua y final
- ▶ Rubricas de evaluación
- ▶ Actividades y observación

# Evaluación

Criterios de calificación: 4 g – 4i, 7a - 7j

- g) Se han definido y utilizado clases heredadas.
- h) Se han creado y utilizado métodos estáticos.
- i) Se han creado y utilizado conjuntos y librerías de clases.

RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

- a) Se han identificado los conceptos de herencia, superclase y subclase.
- b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
- c) Se ha reconocido la incidencia de los constructores en la herencia.
- d) Se han creado clases heredadas que sobrescriben la implementación de métodos de la superclase.
- e) Se han diseñado y aplicado jerarquías de clases.
- f) Se han probado y depurado las jerarquías de clases.
- g) Se han realizado programas que implementen y utilicen jerarquías de clases.
- h) Se ha comentado y documentado el código.
- i) Se han identificado y evaluado los escenarios de uso de interfaces.
- j) Se han identificado y evaluado los escenarios de utilización de la herencia y la composición.

# Evaluación

- ▶ Actitud/trabajo diario 10%.
- ▶ Proyecto 30%.
- ▶ Prueba final 20%.
- ▶ Pruebas y actividades 40%.

# Atención a la diversidad

- ▶ Sesiones de refuerzo y ampliación
- ▶ Adaptaciones en acceso y tiempo según la normativa



# Recursos y bibliografía

- ▶ Usaremos el ordenador, proyector, pizarra, documentación y material de elaboración propia que el alumnado tendrá disponible en aulas.
- ▶ Además, usaremos un IDE, en este caso eclipse y un procesador de textos (LibreOffice Writer)

# Actividad 1

- ▶ **Actividad: Sistema de transporte público con jerarquía de clases**
- ▶ **Contexto laboral:**
  - ▶ Una empresa tecnológica valenciana desarrolla software para la gestión de diferentes medios de transporte público en la provincia. Te asignan la tarea de modelar una jerarquía de clases que represente autobuses, tranvías y metros.
- ▶ **Contenidos trabajados:**
  - ▶ 7.2 Herencia
  - ▶ 7.3 Jerarquía de clases
  - ▶ 7.6 Constructores y herencia
  - ▶ 7.7 Acceso a métodos de la superclase
  - ▶ 7.8 Polimorfismo
- ▶ **Tarea:**
  - ▶ Crea una clase base Transporte con atributos comunes y métodos. Luego crea subclases Autobus, Tramvia, Metro, cada una con un comportamiento distinto en su método informarRuta(). Usa polimorfismo para imprimir rutas distintas.

# Actividad 2

- ▶ **Actividad: Implementación de interfaz para productos exportables**
- ▶ **Contexto laboral:**
  - ▶ Una cooperativa agrícola valenciana quiere identificar qué productos están preparados para exportación. Debes diseñar una interfaz Exportable y aplicarla a las clases de productos según corresponda.
- ▶ **Contenidos trabajados:**
  - ▶ 4.11 Concepto y utilización de interfaces
  - ▶ 7.9 Interfaces
  - ▶ 4.10 Métodos estáticos
- ▶ **Tarea:**
  - ▶ Define una interfaz Exportable con un método informarDestino(). Crea dos clases: Naranja y Aceite, implementando la interfaz y mostrando un mensaje con el país de destino.