

Nombre y apellidos: _____

Fecha y hora de entrega del examen en el aula virtual: _____

Ejercicio 1 (1,75 puntos)

Se desea modelar un sistema de **personajes en un videojuego de aventuras**.

1. Implementa la clase **Personaje**, con:

- **Atributos:** nombre (cadena), habilidad (cadena) y nivel (número entero).
- **Métodos:** toString() y equals(). Dos personajes son iguales si tienen **el mismo nombre y la misma habilidad**.

2. Implementa dos clases que hereden de **Personaje**:

- **Guerrero:**
 - Atributo adicional arma (cadena).
 - Reimplementa toString(). En el método toString() tienes que llamar al de la clase padre y además añadir la información de los atributos propios de la clase.
- **Mago:**
 - Atributo adicional tipoMagia (cadena).
 - Reimplementa toString(). En el método toString() tienes que llamar al de la clase padre y además añadir la información de los atributos propios de la clase.

3. Programa de prueba:

- Crea un **array de personajes** con al menos **dos guerreros y dos magos**.
- Imprime la información de cada personaje.
- Usa equals() para comprobar si dos personajes son iguales.

Ejercicio 2 (1,75 puntos)

Se desea modelar un sistema de **vehículos de transporte**.

1. Implementa la clase **Vehiculo**, con:

- **Atributos:** marca (cadena) y modelo (cadena).
- **Métodos:** toString() y equals(). Dos vehículos son iguales si tienen **la misma marca y el mismo modelo**.

2. Implementa dos clases que hereden de **Vehiculo**:

- **Coche:**
 - Atributo adicional numeroPuertas (número entero).
 - Reimplementa toString(), llamando al método de la clase padre y añadiendo la información propia.

- **Moto:**
 - Atributo adicional cilindrada (número entero).
 - Reimplementa toString(), llamando al método de la clase padre y añadiendo la información propia.

3. Programa de prueba:

- Crea un **array de vehículos** con al menos **un coche y una moto**.
- Imprime la información de cada vehículo.
- Usa equals() para comparar dos vehículos.

Ejercicio 3 (2,5 puntos)

Una plataforma ofrece **cursos online y cursos presenciales**.

1. Implementa la clase Curso, con:

- **Atributos:** titulo (cadena), duracionHoras (número entero), precio (número decimal) y totalCursos (número entero) para contar cuántos cursos se han creado en total.
- **Métodos:** getPrecio(), getNumeroTotalCursos(), toString() y equals(). Dos cursos son iguales si tienen **el mismo título y la misma duración**.

2. Implementa la clase CursoOnline, que hereda de Curso:

- **Atributo adicional:** descuento (número decimal).
- **Métodos:**
 - Sobrescribe getPrecio(), llamando al método de la clase padre y aplicando el descuento.
 - Reimplementa toString(), llamando al método de la clase padre y añadiendo la información propia.

3. Programa de prueba:

- Crea un **array de cursos** con cursos normales y cursos online.
- Muestra el precio de cada curso.
- Usa equals() para comparar dos cursos.
- Muestra el número total de cursos creados.

Ejercicio 4 (4 puntos)

Se desea modelar un sistema de **mensajes en una aplicación de chat**.

1. Implementa la clase abstracta **Mensaje**, con:

- **Atributos:** texto (cadena) y likes (número entero).
- **Métodos:**
 - Método abstracto mostrar().
 - Método darLike() que aumenta en una unidad el número de likes.
 - Métodos toString() y equals(). Dos mensajes son iguales si tienen **el mismo texto y el mismo número de likes**.

2. Implementa la clase **MensajImagen**, que hereda de **Mensaje**.

- **Atributo adicional:** rutaImagen (cadena).
- **Métodos:**
 - Sobrescribe el método mostrar(), que imprimirá el texto y la ruta de la imagen.
 - Reimplementa toString(), llamando al de la clase padre y añadiendo la información propia.

3. Implementa la clase **Contacto**, con:

- **Atributos:** nombre (cadena) y mensajes (array de mensajes, que debe ser dinámico).
- **Métodos:**
 - **Enviar mensaje imagen:** Método que permita que el contacto envíe un mensaje imagen. Este método debe agregar el mensaje recibido al array de mensajes.
 - **Dar like:** Método que permita dar like a un mensaje específico.
 - **Mostrar mensajes:** Método para mostrar todos los mensajes del contacto. Debe mostrar cada mensaje utilizando su método toString().

4. Programa de prueba:

- Crea un **array de contactos**.
- Un contacto envía un mensaje imagen y otro contacto le da like.
- Usa equals() para comparar dos mensajes.