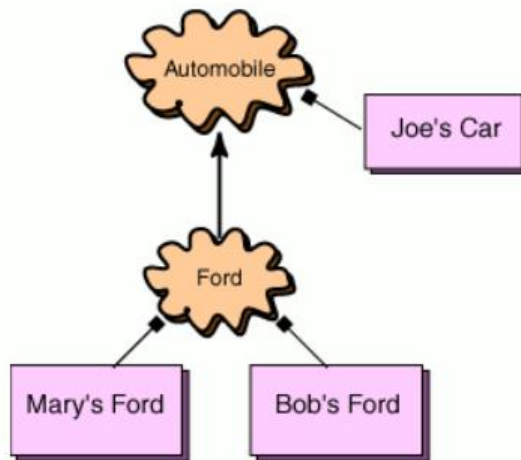


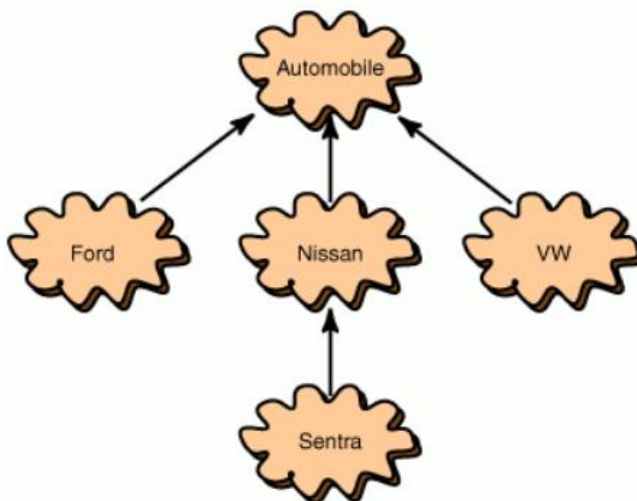
Tarea 1 – Herencia

Enunciado

- En la siguiente figura, se muestra en forma de nube las clases y en forma de rectángulo los objetos que se han construido a partir de las clases. Responde a las preguntas:



- ¿Cuántos tipos de automóviles hay en el diagrama?
 - ¿Cuántos objetos de automóvil hay en el diagrama?
- A partir de la siguiente imagen:

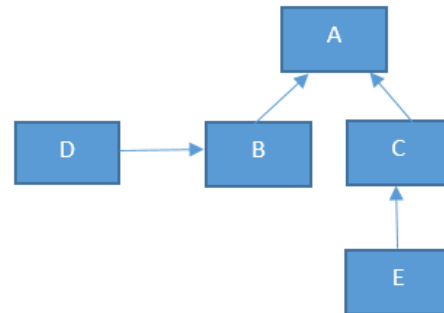


- ¿Es Nissan una clase derivada de Automobile?
 - ¿Es Nissan el padre de Sentra?
 - ¿Cuántos hijos tiene Ford?
 - ¿Sentra heredará los atributos de Automobile?
 - Ford extiende de Automobile. ¿Podríamos tener una clase "Truck" de la que también extienda Ford, es decir, herede de ella? Razona tu respuesta.
- Responde a las siguientes cuestiones.
 - ¿Qué se hereda a nivel de código de una clase base a una clase derivada?

- b. ¿Una clase hija hereda tanto atributos como métodos?
- c. ¿Qué palabra reservada se utiliza para invocar al constructor de la clase padre desde la clase hija?
- d. ¿Un constructor hijo siempre invoca a un constructor padre?
- e. Si la clase padre no tiene constructor definido por el desarrollador y la clase hija si tiene, ¿se generará un error en el constructor de la clase hija si no invocamos al constructor padre? Explica tu respuesta.
4. Pon tres ejemplos de herencia. No hace falta que incluyas atributos o métodos, solo los nombres de las clases y señala quiénes son las clases hijas y las clases padres.
5. Si tenemos una clase padre llamada Video con atributos, título y minutos, y una clase hija llamada Pelicula con atributos como título, minutos, director y valoración. De qué forma habría que implementar las clases para hacer uso de la herencia y reutilizar el mayor código posible. Implementalo.
6. ¿Qué problema surgiría si se permitiese la herencia múltiple en Java? Razona tu respuesta con un ejemplo.

7. A partir del siguiente diagrama:

- a. La clase A es una _____ de la clase C.
- b. La clase B es una _____ de la clase C.
- c. La clase E es una _____ de la clase A.
- d. La clase D es una _____ de la clase B.
- e. La clase A es una _____ de la clase D.



8. Dibuja el diagrama correspondiente a las siguientes clases de herencia: Libro, Novela, Lectura, Periódico.
9. ¿Qué diferencias hay entre sobrecarga y sobreescritura? Pon ejemplo.
10. Crea una clase base llamada 'A' (sin constructor) y una subclase llamada 'B' (sin constructor) que extienda de 'A'. Añade dos atributos en la clase base que hereden a la subclase y un atributo que no herede. Crea tres métodos en la clase padre: dos que se hereden y uno que no. En la clase hija, sobreescríbe uno de los dos métodos que se heredan de la clase padre.
 - a. Si no hemos creado ningún constructor en ninguna clase, ¿qué hace internamente el compilador cuando hacemos: 'B b = new B();'?
 - b. Escribe código para obligar al compilador a usar la palabra super.
11. A partir de las clases generadas en el ejercicio 5. Sobreescríbe el método toString() solo en la clase Video.
 - a. ¿Qué mostrará el siguiente código?
 - b. ¿Por qué no aparecen los atributos de director y valoración en el objeto ob2?

```

public static void main(String[] args) {
    Video ob1 = new Video( titulo: "VHS", minutos: 90 );
    Pelicula ob2 = new Pelicula( titulo: "ET", minutos: 120, director: "Spielberg", valoracion: 10);
    System.out.println(ob1.toString());
    System.out.println(ob2.toString());
}
  
```

UD5. Herencia y polimorfismo

- c. ¿Cómo hay que modificar el código para que también muestre los atributos de director y valoración?
 - d. Implementa una nueva clase llamada `VideoMusical` que herede de `Video`. Tendrá dos variables de instancia: artista y categoría, los dos `String`. Define el constructor y el método `toString`.
 - e. ¿Cuántos atributos o variables de instancia tiene la clase `VideoMusical`? Enuméralos.
 - f. Te habrás dado cuenta que hay un fallo en el diseño, ¡ninguno de los videos tiene precio! Es tu trabajo añadirlo, ¿dónde lo agregarías y qué cambios realizarías en el código?
 - g. ¿Cuál es la clase padre de `Video`?
12. Implementa una clase base llamada `Vehiculo`, luego desarrolla una clase `Coche` que herede de la clase base. Por último, crea una nueva clase que tú quieras y sea específica de `Coche`, es decir, que herede de la clase `Coche`. Habrá que almacenar información sobre su estado y comportamiento (atributos y métodos) como, por ejemplo, nombre, velocidad, marcha, mover, ruedas, etc. Incluye los métodos cambiar de marchas, acelerar/desacelerar velocidad. Tendrás que decidir en qué clase se almacena qué información. Para el tipo más concreto de coche que has definido tendrás que añadir algo específico para ese tipo de coche. No olvides los constructores con todos los parámetros.
13. Escribe una clase con el nombre `Circulo`. La clase necesita un campo (variable de instancia) con `radio` de nombre de tipo doble.
- La clase debe tener un constructor con un parámetro de radio de tipo doble y debe inicializar los campos. En caso de que el parámetro de radio sea menor que 0, debe establecer el valor del campo de radio en 0.
- Escribe los siguientes métodos (métodos de instancia):
- `getRadio` sin ningún parámetro, debe devolver el valor del campo de radio.
 - `getArea` sin ningún parámetro, debe devolver el área calculada ($\text{radio} * \text{radio} * \text{PI}$). Para PI use la constante `Math.PI`.
- Escriba una clase con el nombre `Cilindro` que extienda la clase `Circulo`. La clase necesita un campo (variable de instancia) de nombre `altura` de tipo doble.
- La clase necesita tener un constructor con dos parámetros radio y altura, ambos de tipo doble. En caso de que el parámetro de altura sea inferior a 0, debe establecer el valor del campo de altura en 0.
- Crea los siguientes métodos (métodos de instancia):
- `getAltura` sin ningún parámetro, debe devolver el valor del campo de altura.
 - `getVolume` sin ningún parámetro, debe devolver el volumen calculado. Para calcular el volumen multiplica el área por la altura.
- Crea un método `main` con código de prueba.

Entrega

- Responde y copia el código en un PDF.