Test Examen Ejemplo ISS

1. Es **FALSO**, lo correcto sería al revés (Tren extends MedioTransporte/Autocar extends MedioTransporte).

2. Es **FALSO**, la clase Pasajero no depende de MedioTransporte. La asociación solo indica que MedioTransporte tiene un atributo de clase Pasajero. Además, Pasajero no es una interfaz.

3. Es **FALSO**, pues la clase TresRegional es heredada de Tren y a su vez de MedioTransporte. Por tanto tiene los mismos atributos que Tren y que MedioTransporte.

4. Es **FALSO**, mediante herencia las clases comparten atributos y métodos. La abstracción no está relacionada con la herencia.

5. Es **VERDADERO**, puesto que el atributo «pasajeros» es de clase Pasajero, cada nuevo pasajero que añadamos será un objeto con sus atributos y sus métodos. Utilizando una clase de colección de Java podremos guardar tantos objetos de clase Pasajero como queramos, cada uno con sus características.

6. Es **VERDADERO**, en la "caja" vemos 4 atributos, pero por las asociaciones también vemos que tiene 2 atributos más: un atributo llamado «conductor» de clase Conductor, y un atributo llamado «pasajeros» de clase Pasajero.

7. Es **FALSO,** MedioTransporte tiene un método abstracto que es *frenar()*, pero la clase Autocar tiene su propio método frenar(). El método de MedioTransporte es polimórfico, por lo que cada una de las clases derivadas de MedioTransporte tendrá su propia forma de implementarlo.

8. Es **VERDADERO**, del mismo modo que ocurría con el método *frenar()*, *acelerar()* es un método polimórfico y cada una de las clases derivadas de MedioTransporte lo implementará de una forma diferente.

9. Es **FALSO**, es polimórfica por el mismo motivo que antes.

10. Es **VERDADERO**, en el diagrama solo están definidas clases de tipo Tren o Autocar. Podemos tener distintos tipos de trenes o autocares (Supra, AVE, etc.) pero nunca otros vehículos.

11. Es **FALSO**, los atributos de clase se identifican porque están subrayados.

12. Es **FALSO**, MedioTransporte es una clase abstracta, lo cual significa que su constructor no crea instancias/objetos.

13. Es **VERDADERO**, puesto que MedioTransporte tiene un atributo de clase Conductor.

14. Es **VERDADERO**, en el diagrama vemos que *acelerar()* es una ~~clase~~ (método) abstracta (está en cursiva), pública (+) y de retorno «void».

15. Es **FALSO**, en la clase Tren, el método frenar() **NO** es abstracto.

16. Es **VERDADERO,** la clase Autocar hereda los atributos de MedioTransporte. Por tanto, como MedioTransporte tiene un atributo llamado «conductor» de clase Conductor, en Autocar también lo tenemos.

17. Es **VERDADERO**, la clase AutocarSupra hereda los métodos y atributos de Autocar (extends). Además, depende de Premium, la cual es una interfaz (implements).

18. Es **FALSO**, según el diagrama, las clases Autocar y AutocarSupra no tienen ninguna relación con la clase Azafata. De hecho, la única clase con atributo de tipo Azafata es la clase Ave.

19. Es **FALSO,** el atributo «modeloUniforme» está subrayado, lo cual quiere decir que es un atributo de clase. Esto significa que tomará el mismo valor para todas las instancias que se creen de esa clase.

20. Es **FALSO**, al contrario que en el caso anterior, el atributo «postre» de Menu no es atributo de clase, por lo que cada instancia de Menu podrá tener un atributo «postre» diferente. Lo que podemos observar es que todos los objetos de la clase Menu sí tendrán el mismo precio.