**Autoevaluación UT8**

1. Para declarar **un objeto de una clase** determinada, como **atributo de otra clase**, es necesario especificar que existe una **relación de composición** entre ambas clases mediante el modificador object.
   1. Verdadero
   2. Falso
2. Si se declaran dos variables **objeto a y b de la clase X**, ambas son instanciadas mediante un constructor, y posteriormente se realiza la asignación a=b, el contenido de b será una copia del contenido de a, perdiéndose los valores iniciales de b.
   1. Verdadero
   2. Falso
3. Una **clase derivada hereda** todos los miembros de su **clase base**, pudiendo acceder a cualquiera de ellos en cualquier momento.
   1. Verdadero
   2. Falso
4. Puede **invocarse al constructor de una superclase** mediante el uso de la referencia **this**.
   1. Verdadero
   2. Falso
5. Toda **clase Java** tiene un **método toString** y un método **finalize**
   1. Verdadero
   2. Falso
6. Una **clase abstracta** no podrá ser nunca instanciada
   1. Verdadero
   2. Falso
7. Puede llamarse al **constructor** de una **clase abstracta** mediante el **operador new**
   1. Verdadero
   2. Falso
8. Los métodos de una clase abstracta tienen que ser también abstractos
   1. Verdadero
   2. Falso
9. Los **modificadores final y abstract son excluyentes** en la declaración de un método
   1. Verdadero
   2. Falso
10. Una **interfaz en Java** no puede contener la implementación de un método mientras que una clase abstracta sí.
    1. Verdadero
    2. Falso
11. En Java **una clase no puede heredar** de más de una clase abstracta ni implementar más de una interfaz
    1. Verdadero
    2. Falso
12. Los **métodos de una interfaz** en Java tienen que ser obligatoriamente declarados como **public y abstract**. Si no se indica así, se producirá un error de compilación
    1. Verdadero
    2. Falso
13. ¿Qué **palabra reservada** se utiliza en Java para indicar que **una clase** va a definir los métodos indicados por una **interfaz**?
    1. implements
    2. uses
    3. extends
    4. Los métodos indicados por una interfaz no se definen en las clases pues sólo se pueden utilizar desde la propia interfaz
14. En Java **no está permitida la herencia múltiple** ni para clases ni para interfaces
    1. Verdadero
    2. Falso
15. ¿Cuál de las siguientes características dirías que **NO** es una de las que se suelen considerar como uno de los **tres grandes pilares de la Programación Orientada a Objetos**?
    1. Recursividad
    2. Polimorfismo
    3. Encapsulación
    4. Herencia