**Autoevaluación UT6**

1. Un **objeto y una** clase en realidad hacen referencia al mismo concepto. Podría decirse que **son sinónimos**.
   1. Verdadero
   2. Falso
2. Toda definición de una **clase consta de cabecera y cuerpo**. En la cabecera se definen los atributos de los objetos que se crearán a partir de esa clase y en el cuerpo estarán definidos los distintos métodos disponibles para manipular esos objetos.
   1. Verdadero
   2. Falso
3. Si queremos poder **instanciar objetos** de una clase X desde cualquier parte de un programa, ¿qué modificador o modificadores habrá que utilizar en su declaración?
   1. Private
   2. public
   3. abstract
   4. Ninguno de los anteriores
4. Si disponemos de varios objetos que han sido creados a partir de la misma definición de clase, en realidad tendremos un único objeto, pues hacen referencia a un mismo tipo de clase (plantilla).
   1. Verdadero
   2. Falso
5. Dado que normalmente se pretende encapsular el contenido de un objeto en su interior y permitir el acceso a sus atributos únicamente a través de los métodos, los **atributos** de una clase suelen declararse con el modificador **public**.
   1. Verdadero
   2. Falso
6. ¿Con qué **modificador** puede indicarse en Java que un **atributo es constante**?
   1. Con el modificador constant.
   2. Con el modificador starter.
   3. Con el modificador final.
   4. Con el modificador static
7. ¿Qué elementos forman la **interfaz de un objeto**?
   1. Los atributos del objeto.
   2. Las variables locales de los métodos del objeto.
   3. Los métodos.
   4. Los atributos estáticos de la clase.
8. ¿Con cuál de los siguientes **modificadores** **no** puede ser **declarado** un **método** en Java?
   1. private.
   2. extern.
   3. static.
   4. public.
9. En una clase Java puedes definir tantos métodos con el mismo nombre como desees y sin ningún tipo de restricción pues el lenguaje soporta la sobrecarga de métodos y el compilador sabrá distinguir unos métodos de otros.
   1. Verdadero
   2. Falso
10. La **sobrecarga de operadores** en Java permite “rescribir” el significado de operadores del lenguaje tales como +,-, \*, , etc. Esto puede resultar muy útil a la hora de mejorar la legibilidad del código cuando definimos por ejemplo nuevos objetos matemáticos (números racionales, números complejos, conjuntos, etc.).
    1. Verdadero
    2. Falso
11. La **referencia this** en Java resulta muy útil cuando se quieren utilizar en un método nombres de parámetros que **coinciden** con los **atributos de la clase**.
    1. Verdadero
    2. Falso
12. Si queremos que un **atributo de una clase** sea **accesible** solamente desde el código de la propia clase o de aquellas clases que hereden de ella, ¿qué modificador de acceso deberíamos utilizar?
    1. private.
    2. protected.
    3. public.
    4. Ninguno de los anteriores
13. Los **atributos de una clase** suelen ser declarados como **public** para facilitar el **acceso** y la visibilidad de los **miembros de la clase**.
    1. Verdadero
    2. Falso.
14. Dado que los métodos de una clase forman la interfaz de comunicación de esa clase con otras clases, todos los elementos de una clase deben ser siempre declarados como públicos.
    1. Verdadero
    2. Falso
15. Aunque la **declaración de un objeto** es imprescindible para poder utilizarlo, ese objeto **no existirá** hasta que no se construya una **instancia de la clase del objeto**. Es decir, mientras la clase no sea instanciada, el objeto aún no existirá y lo único que se tendrá será una variable que contendrá un objeto vacío o nulo.
    1. Verdadero
    2. Falso
16. ¿Con qué nombre es conocido el método especial de una clase que se encarga de reservar espacio e inicializar atributos cuando se crea un objeto nuevo? ¿Qué nombre tendrá ese método en la clase?
    1. Método constructor. Su nombre dentro de la clase será constructor..
    2. Método inicializador. Su nombre dentro de la clase será el mismo nombre que tenga la clase..
    3. Método constructor. Su nombre dentro de la clase será el mismo nombre que tenga la clase..
    4. Método constructor. Su nombre dentro de la clase será new.
17. El constructor por defecto (sin parámetros) está siempre disponible para usarlo en cualquier clase.
    1. Verdadero
    2. Falso
18. Toda clase debe incluir un constructor copia en su implementación. ¿Verdadero o falso?
    1. Verdadero
    2. Falso
19. Cuando se abandona el ámbito de un objeto en Java éste es marcado por el recolector de basura para ser destruido. En muchas ocasiones una clase Java no tiene un método destructor, pero si fuera necesario hacerlo ¿podrías implementar un método destructor en una clase Java? ¿Qué nombre habría que ponerle?
    1. Sí es posible. El nombre del método sería finalize().
    2. No es posible disponer de un método destructor en una clase Java.
    3. Sí es posible. El nombre del método sería destructor ().
    4. Sí es posible. El nombre del método sería ~nombreClase, como en el lenguaje C
20. ¿De qué objeto hereda cualquier clase en Java?
    1. Exception
    2. Object
    3. Element
    4. Ninguno de los anteriores
21. La sentencia import nos facilita las cosas a la hora de especificar las clases que queremos utilizar en nuestro archivo Java. Con el uso del comodín (asterisco) podemos importar todas las clases y subpaquetes que se encuentran en un determinado paquete a través de una sola sentencia import.
    1. Verdadero
    2. Falso