

Tipo de examen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Opciones	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Orientación a objetos

1. Unified Modeling Language (UML)
  - a. es una metodología
  - b. es un lenguaje de programación
  - c. es un lenguaje para visualizar, especificar, construir y documentar sistemas informáticos**
  - d. ninguno de los anteriores
2. Cual de las siguientes afirmaciones es cierta con respecto a una clase
  - a. los atributos definen el comportamiento y las operaciones la forma
  - b. los atributos definen la forma y las operaciones el comportamiento**
  - c. tanto los atributos como las operaciones definen la forma
  - d. tanto los atributos como las operaciones definen el comportamiento
3. Los objetos de una misma clase tienen
  - a. la misma forma y comportamiento, pero distintas relaciones
  - b. la misma forma pero distinto comportamiento y distintas relaciones
  - c. la misma forma, el mismo comportamiento y las mismas relaciones**
  - d. las mismas relaciones pero distinta forma y comportamiento
4. Sobre el role
  - a. un objeto de una clase sólo tiene un posible role independientemente de las relaciones en las que intervenga
  - b. un objeto de una clase puede tener diferentes roles, tantos como relaciones en las que intervenga**
  - c. cada objeto de una clase puede tener un role diferente en una misma relación
  - d. ninguna de las anteriores
5. Sobre los diferentes tipos de relaciones
  - a. en asociación y en agregación la semántica está implícita pero en generalización no lo está
  - b. en asociación la semántica no está implícita, y tanto en agregación como en generalización si lo está**
  - c. en asociación la semántica está implícita, y tanto en agregación como en generalización no lo está
  - d. ninguna de las anteriores
6. Sobre la multiplicidad
  - a. en cualquier tipo de relación tiene sentido indicar la multiplicidad
  - b. sólo en asociación se puede indicar la multiplicidad, en los otros tipos de relaciones no tiene sentido

- c. tanto en asociación como en agregación es posible indicar la multiplicidad, pero en generalización no tiene sentido**
  - d. ninguna de las anteriores
7. Una clase abstracta es aquella que
- a. tiene algún método abstracto
  - b. todos sus métodos son abstractos
  - c. no se puede instanciar (al ser un nivel de abstracción demasiado elevado no tiene sentido semánticamente)**
  - d. ninguna de las anteriores
8. Una clase abstracta A tiene únicamente un método abstracto *m()*. La clase A implementa el interface I que declara el método *mi()*. La clase abstracta B extiende a A e implementa el método *mi()*. Si la clase C extiende a B,
- a. C tiene que ser concreta necesariamente e implementar el método *m()*
  - b. C puede ser concreta únicamente si implementa los métodos *m()* y *mi()*
  - c. C puede ser concreta únicamente si implementa el método *m()***
  - d. C puede ser concreta en cualquier caso
9. Si tenemos el interface Movable que presenta la operación *mover()*
- a. todas las clases que realicen el interface deben implementarlo
  - b. todas las clases concretas que realicen el interface deben implementarlo**
  - c. todas las clases abstractas que realicen el interface deben implementarlo
  - d. la operación *mover()* debe estar implementada en el propio interface
10. Sobre las clases y los interfaces
- a. las clases agrupan "cosas" con características comunes y los interfaces definen capacidades o habilidades**
  - b. semánticamente es igual una clase y un interface
  - c. las clases se usan cuando hay herencia simple y los interfaces cuando hay herencia múltiple
  - d. ninguna de las anteriores

## Java

11. Sobre el nombre de las clases
- a. es obligatorio que empiece por mayúscula
  - b. se recomienda que empiece por mayúscula y esté en singular**
  - c. se recomienda que empiece por mayúscula y esté en plural
  - d. no existe ninguna recomendación al respecto
12. Sobre el constructor por defecto
- a. es aquel que pone el compilador cuando no se implementa ningún constructor
  - b. es aquel que no recibe parámetros**
  - c. es aquel que inicializa todos los parámetros con valores por defecto
  - d. puede recibir parámetros y se encarga de inicializar todos los atributos a valores por defecto

13. Se recomienda
  - a. que tanto atributos como métodos sean públicos
  - b. que los atributos sean privados y los métodos que se vayan a usar desde fuera sean públicos**
  - c. que tanto atributos como métodos sean privados
  - d. que los atributos sean públicos y los métodos privados
14. Si tenemos la jerarquía de clases Perro > Mamífero > Animal, cuando se instancia un Perro
  - a. no siempre se invoca a un constructor de Perro, otro de Mamífero y otro de Animal
  - b. siempre se invoca a un constructor de cada superclase y el orden de construcción es Perro, luego Mamífero y finalmente Animal (de lo más específico a lo más genérico)
  - c. siempre se invoca a un constructor de cada superclase sucesivamente, y la construcción siempre se realiza de lo más general a lo más específico (primero Animal, luego Mamífero y finalmente Perro)**
  - d. ninguna e las anteriores
15. La sobrecarga se aplica en métodos con
  - a. mismo nombre, mismos parámetros y mismo retorno
  - b. mismo nombre y distinto retorno
  - c. mismo nombre, distintos parámetros y mismo retorno**
  - d. distinto nombre y mismo retorno
16. La sobrescritura es
  - a. cuando en una clase hay métodos con el mismo nombre
  - b. cuando una subclase reimplementa un método ya implementado en alguna superclase (no tiene porqué estar implementado en la superclase inmediata)**
  - c. cuando una subclase implementa un método definido como abstracto en una superclase
  - d. cuando un método recibe una instancia sin saber de qué clase es y adapta su comportamiento a la misma
17. Si tenemos la clase Alumno que extiende de la clase Persona es correcta la siguiente sentencia
 

```
Persona p = new Alumno();
```

  - a. no, ya que aunque el Alumno es una Persona, Java no lo permite
  - b. no porque el Alumno no es una Persona
  - c. si, aunque Java dará un error si intentamos usar métodos únicamente definidos en la clase Alumno (no definidos a nivel de Persona)**
  - d. si, y Java no dará error aunque intentemos usar métodos no definidos a nivel de Persona
18. Un atributo definido como *private*
  - a. puede ser utilizado en las subclases ya que se hereda
  - b. no puede ser utilizado en las subclases ya que su visibilidad de limita a la propia clase**

- c. a priori no se puede saber si está disponible en las subclases
- d. puede ser utilizado únicamente en el mismo paquete

19. Una clase abstracta

- a. sólo puede contener variables
- b. sólo puede contener constantes
- c. puede contener variables y constantes**
- d. no puede contener ni variables ni constantes

20. Cual de las siguientes afirmaciones es correcta

- a. los interfaces no pueden tener implementación pero si atributos
- b. Java no permite la herencia múltiple pero sí la implementación de múltiples interfaces**
- c. en Java sólo se puede heredar de una única clase y sólo se puede implementar un único interface
- d. los interfaces son exactamente lo mismo que las clases abstractas que tienen todos sus métodos abstractos

21. Cual de las siguientes afirmaciones es correcta a partir del siguiente código:

```
public class Persona
{
    private String nombre;
    private int edad;

    public Persona(String nombre)
    {
        this.nombre = nombre;
        edad = 0;
    }

    public String toString()
    {
        return "[nombre:" + nombre + ",edad:" + edad + "]";
    }

    public static void main(String[] args)
    {
        Persona p = new Persona("Lucas");
        System.out.println( p );
    }
}
```

- a. se produce un error porque no se ha implementado el constructor por defecto
- b. aunque en Java no da error, no se deberían poner los atributos como privados
- c. al ejecutarlo, se imprimirá en pantalla algo similar a "Persona@3e25a5"
- d. al ejecutarlo, se imprimirá en pantalla "[nombre:Lucas,edad:0]"**

22. Cual de las siguientes afirmaciones es correcta a partir del código de las dos clases:

```
public class Coche
{
    private String matricula;
    private String marca;
    private static int coches = 0;

    public Coche(String matricula,String marca)
    {
```

```

        this.matricula = matricula;
        this.marca = marca;
        coches++;
    }

    public static int instancias()
    {
        return coches;
    }
}

public class CocheTest
{
    public static void main(String[] args)
    {
        Coche c1 = new Coche("3948DKS","Seat Panda");
        Coche c2 = new Coche("2938TIE","Renault Clio");
        Coche c3 = new Coche("2938SKD","Honda Accord");
        System.out.println(Coche.instancias());
    }
}

```

- a. Java da error en la última línea del main ya que instancias() es un método de clase que debe ser usado desde una instancia
- b. se imprime en pantalla "3" que es el número de instancias creadas**
- c. Java da error por no haber implementado el constructor por defecto
- d. aunque Java no da error no se debe hacer este uso de los static

23. A partir de las siguientes 3 clases, cual de las afirmaciones es correcta:

```

public class Poligono
{
    public void dibujar()
    {
        System.out.print("POLÍGONO...");
    }
}

public class Circulo extends Poligono
{
    public void dibujar()
    {
        System.out.print("CÍRCULO...");
    }
}

public class TestCirculo
{
    public static void main(String[] args)
    {
        Poligono miPoligono[] = new Poligono[2];
        miPoligono[0] = new Circulo();
        miPoligono[1] = new Poligono();
        miPoligono[0].dibujar();
        miPoligono[1].dibujar();
    }
}

```

- a. el programa imprime por pantalla "POLÍGONO...POLÍGONO..."
- b. el programa imprime por pantalla "CÍRCULO...POLÍGONO..."**
- c. Java da error en la segunda línea del main ya que estamos intentando instanciar un Círculo que hemos definido como Poligono
- d. Java da error en la cuarta línea ya que se produce una ambigüedad (no sabe qué método dibujar() debe ejecutar)

24. A partir del siguiente código, cual de las afirmaciones es correcta:

```
public class Animal
{
    public abstract void comer();
}
```

- a. en principio Java no da error, aunque las subclases de Animal que no sean abstractas deberán implementar el método comer()
- b. Java da un warning ya que un método abstracto no puede devolver void
- c. Java da error ya que si existe un método abstracto, la clase debe ser abstracta**
- d. Java da error ya que un método abstracto no puede ser public

25. A partir del siguiente código (una clase y un interface), cual de las afirmaciones es correcta:

```
public interface Pintable
{
    public void pintar(String nuevoColor);
}

public class Casa implements Pintable
{
    private String dirección;
    private String color;

    public Casa(String direccion)
    {
        this.dirección = direccion;
        this.color = "Blanco";
    }

    public void pintar(String nuevoColor)
    {
        this.color = nuevoColor;
    }
}
```

- a. cualquier clase que extienda a Casa deberá implementar el método pintar() definido en el interface Pintable
- b. una clase que extienda Casa podría ser abstracta**
- c. Java da error ya que la clase Casa debe ser abstracta
- d. Java da error ya que en un interface no se puede implementar ningún método