



Programación de Sistemas
Grado en Ingeniería en Tecnologías de Telecomunicación

Leganés, 12 de marzo de 2019
Duración de la prueba: 20 min

Examen parcial 1 (teoría)
Puntuación: 3 puntos sobre 10 del examen

Sólo una opción es correcta en cada pregunta. Cada respuesta correcta suma 0,3 puntos. Cada respuesta incorrecta resta 0,1 puntos. Las preguntas no contestadas no suman ni restan puntos.

Marca:  Anula:  No uses:   

- Marca la respuesta a cada pregunta con una equis (“X”) en la tabla de abajo.
- Si marcas más de una opción o ninguna opción, la pregunta se considera no contestada.
- Rellena **tus datos personales** antes de comenzar a realizar el examen.

Nombre:

Grupo:

Firma:

NIA:

--	--	--	--	--	--	--	--	--

	A	B	C	D		A	B	C	D
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



1.- Dada la siguiente declaración de atributo en una clase: *int elementos*; ¿Cuál de las siguientes afirmaciones es la correcta?

- (a) *** Si no se asigna valor al atributo java lo inicializará con valor 0.
- (b) *elementos = 10.0f* asignará el valor 10 entero a la variable *elementos*.
- (c) *elementos.length* contiene el número de elementos de la variable.
- (d) *elementos.valueOf("13")* devuelve el int con valor 13.

2.- Dada la siguiente clase. Para llamar del constructor *Coche(String nombre, String marca)* al constructor *Coche(String nombre)* que habría que hacer:

```
Public class Coche{
    private String nombre; private String marca;
    Coche(String nombre) { this.nombre = nombre; }
    Coche(String nombre, String marca){ this.marca = marca; }
}
```

- (a) *** Debe utilizarse *this(nombre)*; como primera sentencia del constructor.
- (b) Debe utilizarse *super(nombre)* como primera sentencia del constructor.
- (c) Debe utilizarse *this(nombre)* implementado en cualquier lugar del constructor.
- (d) Debe utilizarse *super(nombre)* implementado en cualquier lugar del constructor.

3.- Dado el siguiente código. ¿Cuál de las siguientes afirmaciones es la correcta?

```
public class A{
    public static void main(String [] args){ B b = new B(); }
}

public class B extends A { }
```

- (a) *** El código compila y ejecuta creando un objeto de la clase *B* en la referencia *b* aunque no hay constructores implementados.
- (b) El código no compila.
- (c) Es necesario implementar un constructor en la clase *A* y *B* para asegurar la inicialización del objeto.
- (d) Desde la clase *A* no podemos crear instancias de la clase *B*.

4.- Dado el siguiente código, ¿Cuál de las siguientes afirmaciones es correcta al crear una instancia de la clase *B*?:

```
public class A{
    int a;
    public A(int a) { this.a = a;}
```

```

        public static void main(String [] args){
            B b = new B();
        }
    }

```

```

public class B extends A { }

```

- (a) *** El programa no compila generando el compilador un error tipo: *error: constructor A in class A cannot be applied to given types.*
- (b) La clase *B* debería ser privada para poder crear instancias de la clase *B*.
- (c) Se puede instanciar un objeto de la clase *B* puesto que la clase *A* tiene un constructor implementado.
- (d) Es necesario implementar un constructor sin parámetros en la clase *B*.

5.- El polimorfismo las llamadas a los métodos de objetos se resuelven en:

- (a) *** Tiempo de ejecución.
- (b) Tiempo de programación.
- (c) Tiempo de compilación.
- (d) Ninguna de las anteriores es correcta.

6.- Dado el siguiente código, ¿cuál de las siguientes afirmaciones es la correcta?

```

public class Tweet{
    public String mensaje;
    public Tweet() {mensaje="";};
    public void imprimir() { System.out.println(mensaje);}

    public static void main (String [] args){
        Tweet t = new Tweet();
        t.imprimir();
    }
}

```

- (a) *** Esta clase ofrece un pobre nivel de reutilización y un difícil mantenimiento posterior.
- (b) No inicializa el estado del objeto.
- (c) Es necesario declarar la clase como abstracta porque es una generalización.
- (d) La clase está perfectamente implementada en base a los fundamentos de la POO.

7.- Si el compilador de javac nos dice *A.java:3: error: A cannot be instantiated* cuando estoy intentando hacer esto: *A a = new A();* entonces:

- (a) *** La clase *A* es abstracta.
- (b) El objeto *a* es abstracto.

- (c) *a* debe implementar la interfaz *A*.
- (d) Hay que llamar a un constructor con parámetros.

8.- Si todos los objetos en Java tienen la capacidad de clonarse. ¿Cómo implementaría este comportamiento en aquellos objetos que lo necesiten?

- (a) *** Implementando una interfaz *Clonable* que defina un método público *clone()* donde todas las clases que requieran dicho comportamiento implementen la interfaz.
- (b) Con una clase abstracta *Clone* con un método público *clone()* abstracto, donde todas las clases que requieran dicho comportamiento hereden de *Clone* e implementan dicho método.
- (c) Implementando un método público *clone()* en aquellas clases que tengan la capacidad de clonarse.
- (d) Implementando un clase *Clone* y un método público estático *clone()* con el que se realice dicha operación con independencia del objeto.

9.- Respecto a las clases de equivalencia ¿cuál de las siguientes afirmaciones es la correcta?

- (a) *** Se prueba también en los valores frontera de los subconjuntos de valores.
- (b) Las clases de equivalencia permiten generar clases abstractas.
- (c) Son instanciadas por JUnit de manera automática.
- (d) Se prueba con uno de los valores frontera de los subconjuntos de valores.

10.- Respecto a las herramientas de testing ¿cuál de las siguientes afirmaciones es la correcta?

- (a) *** EclEmma nos ayuda en la etapa de pruebas de caja blanca.
- (b) EclEmma nos ayuda en la etapa de pruebas de caja negra.
- (c) EclEmma nos ayuda en la etapa de pruebas de rendimiento.
- (d) EclEmma nos ayuda en la etapa de pruebas de seguridad.