

## Examen de Java

Nombre: \_\_\_\_\_

DNI: \_\_\_\_\_ Titulación: \_\_\_\_\_

1. ¿Cómo podemos compilar desde la línea de comandos la clase Java `A` que utiliza una librería empaquetada `bd.jar`?

2. Indica si es correcto el siguiente código. Justifica la respuesta.

```
List<int> lista = new LinkedList<int>();  
for (int i = 0; i<10 ; i++)  
    lista.add(i,i);
```

3. ¿Sería correcto el siguiente código? Justifica la respuesta.

```
class Uno{  
    protected Uno yoMismo(){ return this;}  
}  
  
class Dos extends Uno{  
    public Dos yoMismo(){  
        return super.yoMismo();  
    }  
}
```

4. Indica si el siguiente enunciado es verdadero o falso, justificando la respuesta: “La construcción de un array cuyo tipo es una clase construye los componentes del array llamando al constructor por defecto de la clase”.

5. Indica los posibles errores del siguiente código justificando el motivo:

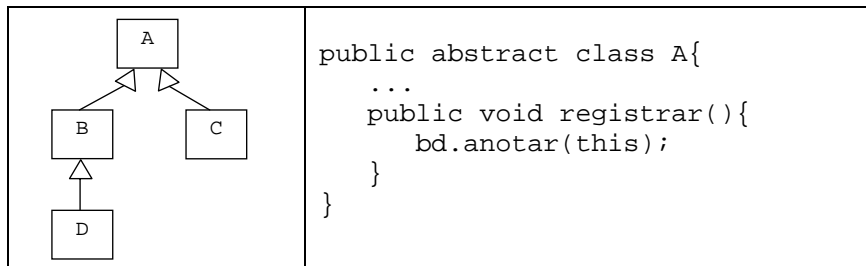
```
public class A{
    private final int at = 5;

    public static int met(){
        return at;
    }
    public void met(){
        System.out.println(at++);
    }
}
```

6. Al aplicar el método `comenzar` sobre la clase `PruebaContador`. ¿Cuál será el valor del atributo `valor` del contador referenciado por `c`? Justifica la respuesta.

|  |  |
|--|--|
| <pre>1. public class PruebaContador{ 2.     private Contador c; 3.     ... 4.     public void comenzar(){ 5.         c = new Contador(1); 6.         this.setContador(c); 7.     } 8.     public void setContador(Contador c){ 9.         c = null; 10.        c = new Contador(3); 11.    } 12. }</pre> | <pre>public class Contador{     private int valor;      public Contador(int v){         valor = v;     }     ... }</pre> |
|--|--|

7. Dada la jerarquía de clases de la figura, ¿cuál será el tipo dinámico de `this` en el método `registrar`? Justifica la respuesta.



8. ¿Cuál es el resultado de la ejecución del siguiente código? Justifica la respuesta.

```
public class Padre{
    public Padre(){
        System.out.println("Padre");
    }
}

public class Hija extends Padre{
    public static void main (String[] args){
        new Padre();
        new Hija();
    }
}
```

9. Indica si el siguiente enunciado es verdadero o falso, justificando la respuesta: "Sea `B` una clase que hereda de `A`, si al llamar a un método heredado se omite la palabra clave `super`, el compilador la inserta automáticamente: `getPlazoDias()` → `super.getPlazoDias()`".

10. Señala los posibles errores del código siguiente y explica por qué lo son.

```
public interface IA{
    int met();
    void met(int i);
}

class A implements IA{
    private int at;
    int met(){
        met(at);
        return at;
    }
}
```

11. Dada la clase `Contenedor` que se muestra a continuación, ¿es correcta la implementación del método `getCopiaContenido`? Dados dos contenedores `c1`, `c2` ¿cuál sería el resultado del mensaje `c1.isIgualContenido(c2)`? Justifica las respuestas.

```
package utilidades;

public class Contenedor{
    private Object contenido;

    public Contenedor (Object contenido){
        if (contenido == null)
            throw new IllegalArgumentException("contenido nulo");
        this.contenido = contenido;
    }
    public Object getCopiaContenido(){
        return contenido.clone();
    }
    public boolean isIgualContenido (Contenedor otro){
        return contenido.equals(otro.contenido);
    }
}
```

12. Dado el siguiente código, ¿es correcta la definición del enumerado? ¿Cuál sería el resultado de la ejecución del método `main` de la clase `Examen`? Justifica las respuestas.

```
public enum Calificacion {
    NP, APROBADO, SUSPENSO;

    public Calificacion getCalificacion(double nota){
        if (nota >= 0 && nota <5) return SUSPENSO;
        else if (nota > 5) return APROBADO;
        else return NP;
    }
}

public class Examen{
    private String alumno;
    private Calificacion nota;

    public Examen(String alumno){
        this.alumno = alumno;
    }
    public void setNota(double n){
        nota = nota.getCalificacion(n);
    }

    public static void main(String[] args) {
        Examen e = new Examen("pepito");
        e.setNota(5.5);
    }
}
```

13. La clase `java.util.TreeSet<T>` forma parte de la librería de colecciones de Java. ¿Existe alguna restricción para su uso cuando se construye utilizando un constructor sin argumentos? Justifica la respuesta.

14. ¿Sería correcto el siguiente código? Justifica la respuesta.

```
public class Plantilla<T extends Empleado>{
    private List<T> empleados;

    public void add(T empleado){
        empleados.add(empleado);
    }

    ...

    public static void main (String[] args){
        Plantilla plantilla = new Plantilla();
        plantilla.add("pepito");
    }
}
```

15. ¿Cuál será el resultado de la ejecución del método `main`? Justifica la respuesta.

```
public class HiloEjecucion{
    public static boolean[] datos = new boolean[3];

    public static String met(int i, boolean valor){
        String salida = "";
        try{
            salida += datos[i];
            datos[i] = valor;
            salida += "OK";
        } catch (Exception e){
            salida += "Excepcion ";
        }
        finally{
            salida += "Finally ";
        }
        salida += "--";
        return salida;
    }

    public static void main(String [] args){
        System.out.println(met(0, true));
        System.out.println(met(3, false));
    }
}
```