

## Preguntas examen

### **1.- Paradigmas de programación en Java:**

- Funcional
- Reactiva
- Programación orientada a objetos

### **2.- Componentes de la programación orientada a objetos y describirlos**

- Variable de referencia: Variable que contiene la dirección de un objeto en memoria. Es un contenedor en la memoria stack donde se guarda la dirección del objeto que está en la memoria heap.
- Referencia: Dirección del objeto en la memoria heap.
- Objeto: instancia de una clase. Se instancian con la palabra clave new.

### **3.- Diferencia entre String y StringBuilder**

String es inmutable (no cambia su valor), tiene un pool de String que contiene varios objetos de tipo String y cambia el comportamiento de equals (ahora compara similitud de cadenas de texto).

StringBuilder es mutable (cambia su valor), no contiene un pool de string crea un objeto individual por string.

### **4.- Explica que es null**

Significa que una variable de referencia no está apuntando a ningún objeto en memoria

### **5.- ¿Cuál es el constructor default?**

Un constructor que implementa java en las clases si es que no colocaste ninguno. Es un constructor sin parámetros.

### **6.- ¿Tipos de variables y sus propiedades?**

- Variable Local: son variables definidas dentro de un método, no se inicializan de forma automática. Solo viven dentro del método.
- Variable de Instancias de clase u objeto, se declaran en el cuerpo de la clase o se instancian en una clase, de no darles un valor se inicializan siempre con null si son objetos, y los primitivos con el valor según les corresponda. Viven mientras sean referenciadas o se las lleva el recolector de basura.
- Variable De clase: son variables estáticas con la palabra reservada Static, se inicializan si son objetos con null y si son primitivos según el valor que les corresponda. Viven mientras se ejecute el programa.

## Preguntas practicas

## Capítulo 2:

### -Pregunta 2

2. What lines are printed by the following program? (Choose all that apply.)

```
1: public class WaterBottle {
2:     private String brand;
3:     private boolean empty;
4:     public static float code;
5:     public static void main(String[] args) {
6:         WaterBottle wb = new WaterBottle();
7:         System.out.println("Empty = " + wb.empty);
8:         System.out.println("Brand = " + wb.brand);
9:         System.out.println("Code = " + code);
10:    }
```

- A. Line 8 generates a compiler error.
- B. Line 9 generates a compiler error.
- C. Empty =
- D. Empty = false
- E. Brand =
- F. Brand = null
- G. Code = 0.0
- H. Code = 0f

**Respuesta correcta:** Inciso D, F y G

Las tres variables: Brand, empty y code son variables de instancia de clase. Siempre se inicializan con sus valores por defecto.

En el caso de Brand se inicializa con null por ser un objeto de tipo String; Por otro lado, Empty y code son datos primitivos, Empty se inicializa en false por ser un boolean y Code se inicializa en 0.0 por ser un float.

### -Pregunta 10:

10. Which statements about the following class are correct? (Choose all that apply.)

```
1: public class PoliceBox {
2:     String color;
3:     long age;
4:     public void PoliceBox() {
5:         color = "blue";
6:         age = 1200;
7:     }
8:     public static void main(String []time) {
9:         var p = new PoliceBox();
10:        var q = new PoliceBox();
11:        p.color = "green";
12:        p.age = 1400;
13:        p = q;
14:        System.out.println("Q1="+q.color);
15:        System.out.println("Q2="+q.age);
16:        System.out.println("P1="+p.color);
17:        System.out.println("P2="+p.age);
18:    } }
```

- A. It prints Q1=blue.
- B. It prints Q2=1200.
- C. It prints P1=null.
- D. It prints P2=1400.
- E. Line 4 does not compile.
- F. Line 12 does not compile.
- G. Line 13 does not compile.
- H. None of the above

**Respuesta correcta:** Inciso C

Línea 4: El método PoliceBox(), a pesar de tener la estructura de un constructor, no funciona como constructor ya que tiene un void. Java lo tomara como un método de la clase y no como constructor. Por esta razón los objetos generados no podrán tomar los valores que contiene.

Java utilizara el constructor por defecto.

En la línea 11 y 12 se establece el color y la edad de "p", sin embargo, en la línea 13, "p" cambia de referencia, ahora apunta al objeto que referencia "q".

Al objeto que referencia “q”, no se le ha dado un color o una edad, sus atributos están vacíos o en 0.

Por lo que al mostrar sus atributos tanto de “q” o “p” en consola, su color será null y su edad 0.

#### Capítulo 4:

##### -Pregunta 6

6. Which statements, when inserted independently into the following blank, will cause the code to print 2 at runtime? (Choose all that apply.)

```
int count = 0;
BUNNY: for(int row = 1; row <=3; row++)
    RABBIT: for(int col = 0; col <3 ; col++) {
        if((col + row) % 2 == 0)
            _____;
        count++;
    }
System.out.println(count);
```

- A. break BUNNY
- B. break RABBIT •
- C. continue BUNNY
- D. continue RABBIT
- E. break
- F. continue
- G. None of the above, as the code contains a compiler error

**Respuesta correcta:** Inciso B, C y E.

Break Rabbit, break y continue Bunny pueden tomarse como la misma declaración ya que las tres rompen o se salen del ciclo Rabbit.

El ciclo de row comenzara iterando desde 1 y el ciclo col desde el 0. Count aumentara cuando la suma de col y row sea diferente de 2. Aumenta por primera vez al ser row = 1 y col = 0 cuando se coloca una de las declaraciones y posteriormente en row= 2 y col = 0. Al colocar una de estas 3 declaraciones en la segunda iteración count valdrá 2.

##### -Pregunta 14:

14. What is the output of the following code snippet?

```
2: double iguana = 0;
3: do {
4:     int snake = 1;
5:     System.out.print(snake++ + " ");
6:     iguana--;
7: } while (snake <= 5);
8: System.out.println(iguana);
```

- A. 1 2 3 4 -4.0
- B. 1 2 3 4 -5.0
- C. 1 2 3 4 5 -4.0
- D. 0 1 2 3 4 5 -5.0
- E. The code does not compile.
- F. The code compiles but produces an infinite loop at runtime.
- G. None of the above

**Respuesta:** El inciso E

El código no compila. La variable snake se colocó dentro del ciclo do while, para que compile debería estar fuera del ciclo.