

# Academia Java

## Quiz 1: Tipos & Clases

### Tipos

1. Qué imprime el siguiente código?

```
int a = -1;
a=(int)(byte)a;
System.out.println(a);
```

2. Qué imprime el siguiente código? Es diferente a la salida del código en la pregunta 1? Justifique.

```
int a = -1;
a=(int)(char)a;
System.out.println(a);
```

3. Qué imprime el siguiente código?

```
float f = 1/10;
System.out.println(f*10 == 1);
```

4. Considere el siguiente fragmento de código, donde foo es de tipo boolean. En algunas ocasiones el programa falla en la última llamada. ¿Qué error genera? ¿Qué valor de foo puede causar problemas?

```
public static Boolean isPositive(double d) {
    Boolean result = null;
    if (d >= 0)
        result = true;
    else if (d < 0)
        result = false;
    return result;
}
```

```
boolean pos = isPositive(foo);
```

5. Qué imprime el siguiente código?

```
char x = '0';
int i = 0;
System.out.println(true ? x : 0);
System.out.println(false ? i : x);
```

6. Qué imprime el siguiente código? Es la salida esperada? Si no lo fuera, cómo se podría arreglar?

```
final long MICROS_PER_DAY = 24 * 60 * 60 * 1000 * 1000;
final long MILLIS_PER_DAY = 24 * 60 * 60 * 1000;
System.out.println(MICROS_PER_DAY / MILLIS_PER_DAY);
```

7. Qué imprime el siguiente código?

```
System.out.println(2.00 - 1.10);
```

8. Considere las dos asignaciones de x. Son equivalentes? Qué pasa en cada caso?

```
x += i  
x = x + i
```

Caso 1:

```
byte x = 0;  
int i=128;
```

Caso 2:

```
Object x = "Que ";  
String i = "interesante!!";
```

## Clases

1. Considere las clases ClaseA, ClaseB, ClaseC, ClaseD y el siguiente método. Cuáles son las posibles clases que pueden instanciar a var? Qué salida por daría por pantalla este método para cada posible instanciación?

```
public void ejercicio1(ClaseA var){  
    System.out.println(var.metodo1());  
    System.out.println(var.metodo2());  
    System.out.println(var.metodo3());  
}
```

2. Considere las clases ClaseA, ClaseB, ClaseC, ClaseD, ClaseE y la interfaz Interfacel. Conteste las siguientes preguntas.

2.a. Se desea crear una clase ClaseX que extienda la clase ClaseA y solo implemente un método con la siguiente signature “public String metodo3(long dato)”. Es posible hacer esto? Cómo? Esto redefine el método metodo3 definido en ClaseA?

2.b. Dada una variable definida de la siguiente forma “ClaseA var = new ClaseD();” que retornan las siguientes instrucciones:

- a. `var instanceof ClaseA`
- b. `var instanceof ClaseB`
- c. `var instanceof ClaseC`
- d. `var instanceof ClaseD`
- e. `var instanceof ClaseE`

2.c. Dada una variable definida de la siguiente forma “Interface var = new ClaseE();” que retornan las siguientes instrucciones:

- a. `var instanceof ClaseA`
- b. `var instanceof ClaseB`
- c. `var instanceof ClaseC`
- d. `var instanceof ClaseD`
- e. `var instanceof ClaseE`

3. Considere la siguiente clase:

```

public class EqExample{
    private int field;
    public EqExample(int init){
        this.field = init;
    }
    public boolean equals(Object o){
        if (!(o instanceof EqExample))
            return false;
        return ((EqExample)o).field==this.field;
    }
}

```

```

EqExample v1 = new EqExample(10);
EqExample v2 = new EqExample(10);

```

3.a. Qué retorna v1==v2?

3.b. Qué retorna v1.equals(v2)?

4. Considere los siguientes métodos main. Qué imprimen?

4.a.

```

public static void main(String []args){
    Integer a = new Integer(10);
    Integer b = new Integer(10);
    System.out.println(a==b);
    System.out.println(a.equals(b));
}

```

4.b.

```

public static void main(String []args){
    Integer a = 10;
    Integer b = 10;
    System.out.println(a==b);
    System.out.println(a.equals(b));
}

```

4.c.

```

public static void main(String []args){
    Integer a = 128;
    Integer b = 128;
    System.out.println(a==b);
    System.out.println(a.equals(b));
}

```

4.d.

```

public static void main(String []args){
    Integer a = -129;
    Integer b = -129;
    System.out.println(a==b);
    System.out.println(a.equals(b));
}

```

4.e. Existe alguna diferencia entre las salidas? Justifique. En los últimos dos casos. Qué pasa cuando se usa 127 o -128?

5. Considere el siguiente código.

```
ArrayList<Integer> al = new ArrayList<>();  
al.add(10);  
al.add(9);  
al.add(8);  
al.add(7);  
al.add(5);  
al.add(4);  
al.add(3);  
al.add(2);  
al.add(1);  
al.add(0);  
al.remove(2);
```

5.a. Cómo queda conformada la lista?

5.b. Qué sucede si se reemplaza la instrucción “al.remove(2);” por “al.remove(Integer.valueOf(2));”?

9.Cuál es la salida del siguiente programa?

```
public class Test{  
    public static void main (String args[]){  
        String str = NULL;  
        System.out.println(str);  
    }  
}
```

- a. NULL
- b.
- c. Error de compilación.
- d. El código ejecuta pero no hay salida.
- e. Error en tiempo de ejecución.