

Examen 1 – Armando Molina Rincón

1. Paradigmas que maneja Java

Orientada a Objetos (POO)

Functional

Reactiva

2. Componentes básicos de POO y explicación

Variables de referencia. Variable que apunta a un objeto en memoria.

Referencia. Almacena la dirección en memoria de un objeto. Relaciona un objeto con su variable de referencia.

Objetos. Instancia de una clase. Creados con la palabra reservada *new*.

3. Diferencia entre String y StringBuilder

Los String son inmutables, se almacenan en un pool de Strings para reutilizar los objetos String con cadenas iguales. Cuando “cambiamos un String” en realidad estamos creando un objeto nuevo y apuntamos a él.

String sobrescribe el método toString para comparar entre contenidos de Strings.

StringBuilder sí puede modificarse (es Mutable). No sobrescribe el método toString, por lo que tiene el mismo comportamiento que el operados ==.

4. Significado de null

Quiere decir que una variable de referencia no apunta a ningún Objeto.

5. Constructor default.

El constructor por default es colocado por el compilador cuando nosotros no definimos explícitamente el constructor de una clase. El constructor por default no recibe ningún parámetro.

En el momento que colocamos un constructor, el constructor por default no es colocado por el compilador.

6. Menciona los tipos de variables y sus propiedades.

Variable local. Son variables declaradas dentro de métodos. No se inicializan automáticamente.

Variables de instancia de clase. Pertenecen a los Objetos (instancias de clase). Cuando no se inicializan toman el valor *null* para los objetos, en cambio, para las variables primitivas se asigna un valor según su tipo de dato.

Variables de clase. Pertenecen a la clase, son declaradas utilizando la palabra reservada *Static*. Al pertenecer a la clase, cuando se modifica, también cambia para todas las instancias de esta clase.

Cuando no se inicializan toman el valor *null* para los objetos, en cambio, para las variables primitivas se asigna un valor según su tipo de dato.

7. Elabora un programa donde implementes Polimorfismo, en una versión con Clases Abstractas y otra versión con Interfaces.

Capítulo 2.2

Líneas impresas por el programa:

- D. Empty = false
- F. Brand = null
- G. Code = 0.0

Explicación: Al ser variables de instancia (propiedades), se inicializan implícitamente con un valor.

- Para Empty de tipo boolean el valor por defecto es false.
- Para Brand de tipo String el valor por defecto es null al tratarse de un Objeto.
- Para Code de tipo float el valor por defecto es 0.0.

Capítulo 2.10

Declaraciones correctas:

- C. P1 = null.

Explicación. El método *public void PoliceBox()* NO es el constructor de la clase por tener la palabra *void*, por lo que *PoliceBox* utiliza el constructor por default y no se asignan las valores “blue” y 1200 a *color* y *age*, respectivamente. Por lo tanto, *color = null* y *age = 0*.

En la línea 13: *p = q*; se le indica a la variable de referencia *p* que apunte al objeto que *q* está apuntando.

La salida del programa debería ser:

- Línea 14: Q1=null (por ser de tipo String)
- Línea 15: Q2=0 (por ser de tipo double)
- Línea 16: P1=null (por ser de tipo String)
- Línea 17: P2=0 (por ser de tipo double)

Capítulo 4.6

Declaración correcta para imprimir 2:

B. break RABBIT

C. continue BUNNY

E. break

Explicación.

row comienza en 1, col comienza en 0.

La variable count aumenta cuando la suma de $col+row$ es diferente de 2. Count aumenta por primera vez cuando $row=1$, $col=0$, colocando break RABBIT, la variable count incrementaría nuevamente en $row=2$, $col=0$. Por lo tanto count=2.

La sentencia break RABBIT es equivalente a break por ser el ciclo inmediato, y es equivalente a continue BUNNY por romper el ciclo RABBIT y continuar el ciclo "Padre".

Capítulo 4.14

La salida del programa es:

E. The code does not compile.

Explicación.

El código no compila porque la variable snake que estamos utilizando en la condición de paro del ciclo do while está declarada dentro del mismo ciclo. El scope de snake es el ciclo do while. Para que compile, snake debería estar declarada fuera del ciclo.