## UT1\_PD4

## Ejercicio 1

Dado el siguiente código fuente:

public class IdentifyMyParts {

public static int x = 7;

public int y = 3;

}

a) ¿Cuáles son las variables de clase?

b) ¿Cuáles son las variables de instancia?

¿Cuál es la salida que produce el siguiente código?

IdentifyMyParts a = new IdentifyMyParts();

IdentifyMyParts b = new IdentifyMyParts();

a.y = 5;

b.y = 6;

a.x = 1;

b.x = 2;

System.out.println("a.y = " + a.y);

System.out.println("b.y = " + b.y);

System.out.println("a.x = " + a.x);

System.out.println("b.x = " + b.x);

System.out.println("IdentifyMyParts.x = " + IdentifyMyParts.x);

1. Las variable de clase está declarada con el modificador static, y es la variable x “public static int x = 7”. Esta variable pertenece a la clase, no a instancias individuales, y se comparte entre todas las instancias de la clase. Existe una sola copia de esta variable
2. La variable de instancia es y, declarada como “public int y = 3”. No tiene el modificador static, almacenan sus estados individuales en campo no estáticos. Sus valores son únicos para cada instancia de una clase (para cada objeto, por ejemplo).

La salida del programa es

a.y = 5

b.y = 6

a.x = 2

b.x = 2

1. IdentifyMyParts.x = 2

Esto se debe a que, “y” al ser una variable de instancia, permite que cada objeto tenga su propia copia de dicha variable. Por esto mismo se imprimen los valores asignados para cada objeto respectivamente.

Sin embargo, el caso de la variable “x” es distinto, ya que este al ser estático va a tener una sola copia de él mismo. Por este motivo, todas las referencias apuntan al último valor asignado (b.x=2), lo mismo ocurre al final cuando se accede a la variable de clase directamente.

## Ejercicio 2

En repositorio

## Ejercicio 3

1. Luego de que el código se ha ejecutado, existe para cada uno de los objetos en este caso 1 referencia. Ninguno de los objetos es candidato al garbage collector ya que se encuentran con referencias activas, el único caso posible es asignando null a studentName, ya que se eliminan referencias y se deja el objeto más propenso a ser eliminado por el programa
2. En un programa de Java, los objetos no se destruyen explícitamente, el garbage collector lo hace cuando no quedan referencias hacia ellos. Una forma de facilitar esto es asignando null a objetos, como mencionado anteriormente.