

Módulo Objetos

Objetos y Clases en Java

Objetivo:

- Realizar programas donde se instancien objetos, a partir de clases existentes, y se envíen mensajes a estos objetos.
- Aprender a trabajar con objetos Strings.
- Comprender los conceptos: clase/objeto/estado/método/mensaje/referencia.

1- Se dispone de una clase **Persona (descargar del entorno)**. Un objeto persona puede crearse sin valores iniciales, agregando luego su nombre, DNI y edad (en ese orden). Un objeto persona responde a los siguientes mensajes:

getNombre()	retorna el nombre (String) de la persona
getDNI()	retorna el dni (int) de la persona
getEdad()	retorna la edad (int) de la persona
setNombre(X)	modifica el nombre de la persona al "String" pasado por parámetro (X)
setDNI(X)	modifica el DNI de la persona al "int" pasado por parámetro (X)
setEdad(X)	modifica la edad de la persona al "int" pasado por parámetro (X)
toString()	retorna un String que representa al objeto. Ej: "Mi nombre es Mauro , mi DNI es 11203737 y tengo 70 años"

a) Realice un programa que cree un objeto persona con datos leídos desde teclado. Luego muestre en consola la representación de ese objeto en formato String.

2- Utilizando la clase Persona (ya implementada):

a) Realice un programa que almacene en un vector 15 personas. La información de cada persona debe leerse de teclado.

Luego de almacenar la información:

b) Agregue un método que devuelva la cantidad de personas mayores de 65 años.

c) Agregue un método que devuelva la representación de la persona con menor DNI.

d) Muestre desde el programa principal la cantidad de personas mayores de 65 años y muestre también la representación por pantalla de una persona con menor DNI.

3- Dados los siguientes programas:

<pre>public class Ej03QueImprimeA { public static void main(String[] args) { String saludo1=new String("hola"); String saludo2=new String("hola"); System.out.println(saludo1 == saludo2); System.out.println(saludo1 != saludo2); System.out.println(saludo1.equals(saludo2)); } }</pre>	<pre>public class Ej03QueImprimeB { public static void main(String[] args) { Persona p1; Persona p2; p1 = new Persona(); p1.setNombre("Pablo Sotile"); p1.setDNI(11200413); p1.setEdad(40); p2 = new Persona(); p2.setNombre("Julio Toledo"); p2.setDNI(22433516); p2.setEdad(51); p1 = p2; p1.setEdad(p1.getEdad() + 1); System.out.println(p2.toString()); System.out.println(p1.toString()); System.out.println((p1 == p2)); } }</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- ¿Qué imprime cada uno?
- ¿Qué efecto tiene la asignación utilizada con objetos?
- ¿Qué se puede concluir acerca de la comparación con == y != utilizada con objetos?
- ¿Qué retorna el mensaje equals cuando se le envía a un String?

4- Se dispone de la clase **Partido (descargar del entorno)**. Un objeto partido representa un encuentro entre dos equipos (local y visitante). Un objeto partido puede crearse sin valores iniciales y agregando luego el nombre del equipo local, el nombre del visitante, la cantidad de goles del local y del visitante (en ese orden). Un objeto partido sabe responder a los siguientes mensajes:

getLocal()	retorna el nombre (String) del equipo local
getVisitante()	retorna el nombre (String) del equipo visitante
getGolesLocal()	retorna la cantidad de goles (int) del equipo local
getGolesVisitante()	retorna la cantidad de goles (int) del equipo visitante
setLocal(X)	modifica el nombre del equipo local al "String" pasado por parámetro (X)
setVisitante(X)	modifica el nombre del equipo visitante al "String" pasado por parámetro (X)
setGolesLocal(X)	modifica la cantidad de goles del equipo local "int" pasado por parámetro (X)
setGolesVisitante(X)	modifica la cantidad de goles del equipo visitante "int" pasado por parámetro (X)
hayGanador()	retorna un boolean que indica si hubo (true) o no hubo (false) ganador
getGanador()	retorna el nombre (String) del ganador del partido (si no hubo retorna un String vacío).
hayEmpate()	retorna un boolean que indica si hubo (true) o no hubo (false) empate

a) Implemente un programa que cargue un vector con los 325 partidos disputados en la superliga 2018/2019.

Luego de la carga realizar los métodos que considere necesarios para informar:

- La cantidad de partidos que ganó River.
- El total de goles que realizó Boca jugando de local.
- El porcentaje de partidos finalizados con empate.



5- Realice un programa que cargue un vector con 10 strings leídos desde teclado. El vector generado tiene un mensaje escondido que se forma a partir de la primera letra de cada string. Muestre el mensaje escondido en consola.

NOTA: La primer letra de un string se obtiene enviándole el mensaje *charAt(0)* al objeto string. Probar con: humo oso lejos ala menos usado nene de ocho ! Debería imprimir: holamundo!