

PROGRAMACIÓN ORIENTADA A OBJETOS

TALLER No. 4 JAVA

PROGRAMACIÓN ORIENTADA A OBJETOS

Profesor: Jaime Alberto Guzmán Luna

Contenido del taller:

- 1) Sobrecarga de constructores
- 2) Sobrecarga de métodos
- 3) Destructores

Ejercicio 1 para analizar

```
1 public class ObjTaller4 {
2
3     public static void main(String[] args) {
4         String[] nombres = {"Alejandro", "Jaime", "Daniel",
5 "Santiago", "Alexander"};
6         Persona[] personas = new Persona[5];
7
8         for (int i = 0; i < nombres.length; i++) {
9             personas[i] = new Persona(nombres[i]);
10        }
11        Vehiculo auto = new Vehiculo("ABC-306", Motor.ELECTRICO,
12 personas[4]);
13
14        Vehiculo auto2 = new Vehiculo("TXT-452", Motor.GASOLINA);
15
16        personas[0] = null;
17        personas[1] = personas[2];
18        personas[4] = null;
19        personas[3].finalize();
20        System.out.println(personas[3]);
21    }
```

```
1 public class Vehiculo{
2
3     private String placa;
4     private Persona dueno;
5     private Motor motor;
6
7     public Vehiculo(String placa, Motor motor, Persona persona) {
8         this.dueno = persona;
9         this.placa = placa;
10        this.setMotor(motor);
11    }
```

PROGRAMACIÓN ORIENTADA A OBJETOS

```
12 public Vehiculo(String placa, Motor motor) {
13     this.placa = placa;
14     this.setMotor(motor);
15 }
16
17 public String getPlaca() {
18     return placa;
19 }
20
21 public void setPlaca(String placa) {
22     this.placa = placa;
23 }
24
25 public Persona getDueno() {
26     return dueno;
27 }
28
29 void setDueno(Persona persona) {
30     this.dueno = persona;
31 }
32
33 public Motor getMotor() {
34     return motor;
35 }
36
37 public void setMotor(Motor motor) {
38     this.motor = motor;
39 }
40
41
42 }
```

```
1 public class Persona {
2
3     private String nombre;
4     private Persona mejorAmigo;
5
6     public Persona(String nombre) {
7         this.nombre = nombre;
8     }
9
10    public void finalize() {
11        System.out.println("Matando a: " + nombre);
12    }
13
14    public String getNombre() {
15        return nombre;
16    }
17
18    public void setNombre(String nombre) {
19        this.nombre = nombre;
20    }
21 }
```

PROGRAMACIÓN ORIENTADA A OBJETOS

```
21     }
22
23     public Persona getMejorAmigo() {
24         return mejorAmigo;
25     }
26
27     void setMejorAmigo(Persona mejorAmigo){
28         this.mejorAmigo = mejorAmigo;
29     }
30
31     public String toString() {
32
33         return "Soy "+nombre;
34     }
35 }
36 }
```

```
1  public enum Motor {
2
3      ELECTRICO(100), GASOLINA(90), DIESEL(80), HIBRIDO(70);
4
5      //velocidad maxima de cada tipo de motor
6
7      private int velocidadMaxima;
8
9      //Añadir un constructor
10     private Motor(int s){
11         velocidadMaxima=s;
12     }
13
14     public int getVelocidadMaxima(){
15         return velocidadMaxima;
16     }
17
18     public void setVelocidadMaxima(int velocidadMaxima){
19         this.velocidadMaxima = velocidadMaxima;
20     }
21 }
```

Actividad:

- A. ¿Qué pasaría si se pierde la referencia de un objeto de tipo Persona?
- B. ¿Como podríamos conocer el nombre del dueño de la variable auto de la **línea 11** del método main?
- C. ¿De qué manera podemos agregar un dueño al Vehículo de la **línea 13** del método main?
- D. Usando la variable auto2 de la **línea 13** del método main, obtenga el valor del atributo velocidadMáxima del motor del vehiculo. Adjunte su propuesta
- E. Suponga que, al momento de perder la referencia al objeto, se borra del sistema, es decir el garbage collector es muy eficiente, ¿Que imprimiría al ejecutar el método main por consola?
- F. ¿Qué ocurre al momento de ejecutar la siguiente línea después de la **línea 16** System.out.println(personas[1]) ? Explique

PROGRAMACIÓN ORIENTADA A OBJETOS

- G. ¿Qué modificación al código debo hacer para que al momento de ejecutar `System.out.println(auto2)`, me aparezca la placa del vehículo y el dueño del vehículo?
- H. Usando la variable `auto` de la **línea 11** del método `main`, y usando el atributo `dueno`, asigne de `mejorAmigo` al tercer elemento del listado `personas`. Adjunte su propuesta
-

Ejercicio 2 para analizar

- A. Según el siguiente código, indique qué se imprime por consola y explique el porqué de cada línea donde se imprime

```
1 public class ObjTaller4B {
2
3     public static void main(String[] args) {
4
5         char c = 'g';
6         short s = 2;
7         byte b = 1;
8         long l = 999999999;
9         int i = 51232;
10        double d= 12.4;
11        float f = 5.65f;
12
13        System.out.println("char : " + funcion(c));
14        System.out.println("short : " + funcion(s));
15        System.out.println("byte : " + funcion(b));
16        System.out.println("long : " + funcion(l));
17        System.out.println("integer : " + funcion(i));
18        System.out.println("double : " + funcion(d));
19        System.out.println("float : " + funcion(f));
20    }
21
22    // static String funcion(short a) {
23    //     return "Entra a short: " + a;
24    // }
25    static String funcion (int a) {
26        return "Entra a int: " + a;
27    }
28
29    static String funcion (double a) {
30        return "Entra a double: " + a;
31    }
32
33    // static String funcion (float a) {
34    //     return "Entra a float: " + a;
35    // }
36    }
37
38    /**
39     * byte short char int long float double
40     */
```

PROGRAMACIÓN ORIENTADA A OBJETOS

B. Realice los siguientes cambios, teniendo siempre como referencia el código inicial. Explique cómo y por qué cambia lo que se imprime por pantalla.

Type your text

- Active la función que recibe un short.
- Active la función que recibe un float.
- Comente la función que recibe un double y active la que recibe un float.
- Comente todas las funciones, excepto la que recibe un double.

Ejercicio 3 - GitHub

Enlace entrega y ejercicio: <https://classroom.github.com/a/3MdacUCd>

PROBLEMA

Según el código dentro del paquete classroom identifique los errores de compilación y piense la razón por la que ocurren, luego corrijalos.

IMPORTANTE

Corrija los errores de la forma que mejor lo considere, de modo que el programa funcione correctamente e imprima por pantalla al correrlo el siguiente resultado. Puede agregar, eliminar o modificar el código. Trate de pensar en la mejor forma de corregirlo.

NOTA: No se debe modificar la clase Taller4 que se encuentra dentro del paquete classroom o tendrá una penalización de 0 en la calificación del taller.

Resultado:

```
Sin nombre
10122
882
50
0
1
```

Para pensar:

1. ¿Qué utilidad tiene la sobrecarga de métodos? ¿Por qué no simplemente llamarlos con nombres diferentes?
2. ¿Por qué cree que la firma de un método no está compuesta por el nombre de los argumentos ni el tipo de retorno para diferenciarlo de los demás?
3. ¿Qué ventajas puede encontrar en un inicializador estático frente a simplemente definir el valor inicial de un atributo en la declaración del mismo?
4. ¿Qué ventajas tiene poder llamar a otro constructor con la expresión `this()`?