

Parte 4 - Programacion en Java | POO

Clase 20 | 26/05

Conceptos Aprendidos

En Clase:

- Métodos Constructores Continuación.
- Uso de condicionales para asignar atributos.

Fuera de clase:

- Amplié conocimientos con el curso paralelo que realizo de mooc-fi
- Uso de Ternary
- Mejora de Portafolio.
 - [Uso de links en markDown.](#)

Primera parte

Continuamos con **Programación Orientada a Objetos.**

Seguimos con el ejercicio de ayer, pero ya lo había terminado como tarea. O sea, que aproveche la mañana para aprender un poco más acerca de la *sobrecarga de métodos*. Dejo a continuación una parte del ejercicio. Todo el desarrollo está [aquí](#).

```
package com.nacho;

import java.util.ArrayList;

public class AspiradoraRobotin {
    private String nombre = "Cortana";

    private boolean prendido = false; // Check si está prendido o no
    private int deposito = 0; // Deposito de Polvo
    private int bateria = 100; // Nivel De Batería
    private boolean bateriaOk = true;
    private boolean aspiradoCompleto = false; // Si completó o no el aspirado
    private ArrayList<String> robot = new ArrayList<>(); // Lista de robots a
limpiar

    private String aviso = " *** Aviso ***";
    private String printName = this.nombre + " dice: ";
    // Variable para no escribir tanto. Si, sé que podría usar
    // un getter o toString. Per elegí esto.
```

```

public AspiradoraRobotin() {
}

public AspiradoraRobotin(String nombre) {
    this.nombre = nombre;
}

/* Getters y Setters */
public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getNombre() {
    return nombre;
}

public int getBateria() {
    return bateria;
}

public void setPerimetro(ArrayList<String> perimetro) {
    this.robot = perimetro; // Lista de Robots a Limpiar
}

private boolean checkList(){
    return deposito<100 && bateria > 5;
    // Verdadero para niveles normales, falso en lo contrario.
}

public void encendido() {
    if (!prendido && checkList()) {
        // Check si esta NO prendido y los niveles del deposito
        // y batería son normales.
        System.out.println();
        System.out.println(printName + "Buenas, soy " +
            this.nombre + " y estoy lista para laburar!!");
        System.out.println();
        System.out.println(avisos);
        prendido = true;
        System.out.println(printName + " el nivel de BATERIA es " + bateria);
        System.out.println(printName + ": el nivel del DEPOSITO es " +
deposito);
    } else if (deposito<100) { // deposito de polvo lleno
        System.out.println(printName + "Tengo lleno el deposito. Lavame,
suci@!");
        vaciarDeposito();
    } else { // batería descargada
        System.out.println(printName + "No tengo más batería. Procedo a
cargarme!");
        cargarBateria();
    }
}
}

```

```

public void apagado() { // Apagado
    System.out.println(printName + "Adiós mundo Cruel");
    prendido = false;
}

public void cargarBateria() { //Simple Loop que carga la batería
    System.out.println(avisos);
    System.out.println("Batería Baja");
    while (getBateria() < 100) {
        bateria += 25;
        System.out.println("Cargando: Batería al: " + bateria + " %");
    }
}

public void vaciarDeposito() { // Simple Loop que vacía del deposito
    if (deposito >= 100) {
        System.out.println(avisos);
        System.out.println(printName + " Deposito Lleno");
        System.out.println(printName + " Vacinando Deposito....");
        System.out.println("****");
        System.out.println("***");
        System.out.println("**");
        System.out.println(avisos);
        System.out.println("Deposito Limpio");
        deposito = 0;
    }
}

public void agregarAlPerimetro(String nuevoRobot) {
    this.robot.add(nuevoRobot); // Agrega nuevo robot a la lista
    System.out.println(printName + "Se agregó el nuevo Robot ==> " +
nuevoRobot);
}

public void limpiar() {

    if (prendido) { //Check si está prendido.
        System.out.println(printName + "Los Robots a limpiar : ");
        for (var j : this.robot){
            System.out.println(j); // Listado de Robots a Limpiar
        }
        var i = 0;
        while (i <= this.robot.size() - 1) {
            /*
            Iterando por la lista hasta que todos los robots estén limpios.
            Imprime que robot está limpio.
            Disminuye los niveles de batería y deposito en 25 por cada

```

```

limpieza.
        Remueve los limpios de la lista
        */
        aspiradoCompleto = false;
        if (checkList() ) {
            System.out.println();
            System.out.println(printName + "Estoy aspirando a ==> " +
this.robot.get(i));
            System.out.println();
            bateria -= 25;
            deposito += 25;
            i += 1;
        } else {
            if(getBateria()<5) cargarBateria();
            else vaciarDeposito();
        }
    }
    aspiradoCompleto = true; // Cuando finaliza, es Verdadero.
} else {
    System.out.println(avisos);
    System.out.println("Encender Primero!");
}
if (aspiradoCompleto) {
    /*
    Si el aspirado está completo, se eliminan los elementos de la lista.
    Avisa que se apagará. Ejecuta función de apagado.
    */
    this.robot.clear();
    System.out.println("**** AVISO ****");
    System.out.println(printName + "Aspirado Completo");
    System.out.println("");
    System.out.println("Apagando en ...");
    for (var j = 3; j >= 0; j--) {
        System.out.println(j);
    }
    System.out.println();
    apagado();
}
}
}

```

Segunda Parte

Nos asignaron el siguiente ejercicio:

Crea una clase llamada Contador que contenga un único atributo entero llamado cont. La clase tendrá los siguientes constructores:

- Constructor por defecto

- Constructor con parámetros para inicializar el contador con un valor no negativo.

Si el valor inicial que se recibe es negativo, el contador tomará el valor cero como valor inicial. Además de los métodos getter y setter, la clase contendrá los métodos:

- incrementar: incrementa el contador en una unidad.
- decrementar: decrementa el contador en una unidad.

El contador nunca podrá tener un valor negativo. Si al decrementar se alcanza un valor negativo, el contador toma el valor cero. Una vez creada la clase, escribe un método main para probarla.

Resultado

```
public class Contador {
    private int cont = 0;

    public Contador() {
    }

    public Contador(int cont) {
        if (cont > 0) {
            this.cont = cont;
        }
        else{
            this.cont=0;
        }
    }

    public int getCont() {
        return cont;
    }

    public void setCont(int cont) {this.cont = cont;}

    public void incrementar(){
        this.cont++;
        System.out.println("Contador: " +getCont());
    }

    public void decrementar(){
        if(getCont()<=0){
            this.cont = 0;
        }else{
            this.cont--;
        }
        System.out.println("Contador: " +getCont());
    }
}
```