



ACTIVITAT

Objectius:

- Aprendre a programar objectes amb el patró 'prototype' en JAVA

Instruccions:

- Es tracta d'un treball en grups de dos, no s'admet cap tipus de còpia.
- Responen a l'espai de cada pregunta, si ho feu amb diapositives enganxeu la diapositiva en aquest mateix espai.
- Es valorarà la cura en la presentació del document i que segueixi l'estructura indicada.

Criteris d'avaluació:

- Cada pregunta té el mateix pes sobre 90%
- Les metodologies de treball, organització personal i participació contenen un 10%

Entrega:

- Aquest document amb les explicacions i captures necessàries i els arxius adjunts necessaris del codi que es demana
- El nom dels arxius adjunts a entregar serà: nomicognom-nomicognom.zip

Noms i Cognoms: Patricio André Rojas Condori

Materials:

Necessiteu un entorn de desenvolupament en JAVA

Feu servir Google per buscar els tutorials que us serveixin millor

Tens més informació sobre el mètode prototype en [aquest post](#) i en [aquest exemple](#) JAVA

Creeu els arxius a la carpeta 'src' del projecte i executeu amb els scripts './build.sh' i './build.ps1'



Tasques:

0 - Implementa els següents objectes en JAVA, fent servir el model de clonació prototype i demostra que funciona correctament:

- Una classe Electrodomèstic amb nom, color, preu, marca, eficiència
- Una classe Rentadora que és un electrodomèstic i a més té: revolucions, soroll
- Una classe Nevera que és un electrodomèstic i a més té: frigories, soroll
- Una classe Forn que és un electrodomèstic i a més té: temperatura, autoneteja

A partir de les classes anteriors, crea una llista de instàncies per cada tipus d'electrodomèstic amb almenys 2 elements de cada tipus (Rentadora, Nevera, Forn), inventa't els valors

Crea després una llista on cada objecte és un clon dels objectes de la llista principal anterior

Fes dos bucles:

- A) Compara la llista original amb ella mateixa, compara que són el mateix objecte, de la mateixa classe i tenen iguals dades
 - B) Compara la llista original amb dels clons, compara que són objectes diferents, de la mateixa classe i tenen iguals dades
- Les comparacions han de ser de cada element amb l'equivalent de l'altre llista (és a dir 1 a 1, no 1 a tots)

Resolució:

Aquestes són les classes creades amb les especificacions de l'enunciat:

- La classe abstracta de la qual deriven les altres:

```
public abstract class Electrodomestic {
    public String nom;
    public String color;
    public double preu;
    public String marca;
    public int eficiència;

    public Electrodomestic() {}

    public Electrodomestic(Electrodomestic target) {
        if (target != null) {
            this.nom = target.nom;
            this.color = target.color;
            this.preu = target.preu;
            this.marca = target.marca;
            this.eficiència = target.eficiència;
        }
    }

    public abstract Electrodomestic clone();

    @Override
    public boolean equals(Object object2) {
        if (!(object2 instanceof Electrodomestic))
            return false;
        Electrodomestic cast2 = (Electrodomestic) object2;
        return cast2.nom.equals(nom) && cast2.color.equals(color) && cast2.preu == preu && cast2.marca.equals(marca) && cast2.eficiència == eficiència;
    }
}
```

- Classe Rentadora



```
public class Rentadora extends Electrodomestic{
    public double revolucions;
    public double soroll;

    public Rentadora(){

    };

    public Rentadora(Rentadora target) {
        super(target);
        if (target != null) {
            this.revolucions = target.revolucions;
            this.soroll = target.soroll;
        }
    }

    @Override
    public Electrodomestic clone() {
        return new Rentadora(this);
    }

    @Override
    public boolean equals(Object object2) {
        if (!(object2 instanceof Electrodomestic) || !super.equals(object2)) return false;
        Rentadora cast2 = (Rentadora) object2;
        return cast2.revolucions == revolucions && cast2.soroll == soroll;
    }
}
```

- Classe Nevera



```
public class Nevera extends Electrodomestic{
    public int frigories;
    public int soroll;

    public Nevera() {

    }

    public Nevera(Nevera target) {
        super(target);
        if (target != null) {
            this.frigories = target.frigories;
            this.soroll = target.soroll;
        }
    }

    @Override
    public Electrodomestic clone() {
        return new Nevera(this);
    }

    @Override
    public boolean equals(Object object2) {
        if (!(object2 instanceof Electrodomestic) || !super.equals(object2)) return false;
        Nevera cast2 = (Nevera) object2;
        return cast2.frigories == frigories && cast2.soroll == soroll;
    }
}
```

- Classe Forn



```
public class Forn extends Electrodomestic{
    public int temperatura;
    public boolean autoneteja;

    public Forn() {

    };

    public Forn(Forn target) {
        super(target);
        if (target != null) {
            this.temperatura = target.temperatura;
            this.autoneteja = target.autoneteja;
        }
    }

    @Override
    public Electrodomestic clone() {
        return new Forn(this);
    }

    public boolean equals(Object object2) {
        if (!(object2 instanceof Electrodomestic) || !super.equals(object2)) return false;
        Forn cast2 = (Forn) object2;
        return cast2.temperatura == temperatura && cast2.autoneteja == autoneteja;
    }
}
```

Després farem la clonació i posterior comprovació dels elements de les llistes.



```
        llista.add(102);

        // Clonar la llista
        for (Electrodomestic obj : llista) {
            llistaCopy.add(obj.clone());
        }

        System.out.println("Comparar la mateixa llista:");
        for (int i = 0; i < llista.size(); i++) {
            compare(i, llista.get(i), llista.get(i));
        }

        System.out.println("Comparar amb la llista clonada:");
        for (int i = 0; i < llista.size(); i++) {
            compare(i, llista.get(i), llistaCopy.get(i));
        }
    }

    static void compare (int i, Electrodomestic a, Electrodomestic b) {
        if (a == b) {
            System.out.println(i + ": Els electrodomestics són el mateix objecte");
        } else {
            System.out.print(i + ": Els electrodomestics són objectes diferents - ");
            if (a.equals(b)) {
                System.out.println(i + ": Els electrodomestics són idèntics");
            } else {
                System.out.println(i + ": Els electrodomestics NO són idèntics");
            }
        }
    }
}
```

Aquest seria el output per consola, com es pot veure els elements comparats son diferents objectes però comparteixen els mateixos valors.

```
• Comparar la mateixa llista:
0: Els electrodomestics són el mateix objecte
1: Els electrodomestics són el mateix objecte
2: Els electrodomestics són el mateix objecte
3: Els electrodomestics són el mateix objecte
4: Els electrodomestics són el mateix objecte
5: Els electrodomestics són el mateix objecte
Comparar amb la llista clonada:
0: Els electrodomestics són objectes diferents - 0: Els electrodomestics són idèntics
1: Els electrodomestics són objectes diferents - 1: Els electrodomestics són idèntics
2: Els electrodomestics són objectes diferents - 2: Els electrodomestics són idèntics
3: Els electrodomestics són objectes diferents - 3: Els electrodomestics són idèntics
4: Els electrodomestics són objectes diferents - 4: Els electrodomestics són idèntics
5: Els electrodomestics són objectes diferents - 5: Els electrodomestics són idèntics
patricio@patricio-Modern-14-C12M:~/Documentos/PracticaPrototype/DAM-JavaExemplePrototype-main$
```