CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

UNITAT 8
POO (I)
EXEMPLES

PROGRAMACIÓ
CFGS DAW

Autors:
Joan Vicent Cassany – jv.cassanycoscolla@edu.gva.es
Revisat per:

2022/2023

Llicència

Exemple 01

Programa que crea gossos i descriu el seu comportament.

CLASSE GOS

```
package UF08_Exemple01;
/**
 * UF08 Exemple 01: Classe Gos
 */
public class Gos {

    // Atributs
    String color, raça, sexe;
    int edat;
    double pes;

    // Mètode constructor: sempre té el mateix nom que la classe

    Gos (String s) {
        this.sexe = s;
    }

    // Mètodes getter
    String diMeSexe() {
        return this.sexe;
    }


    // Mètodes

    // El gos lladra
    void lladra() {
        System.out.println("Bof! Bof!");
    }

    // El gos meneja la cua
    void cua() {
        System.out.println("\\ / \\ / \\ /");
    }
```

```
    // Li donem mejar al gos i ens diu si li agrada
    void menjar(String menjar) {
        if (menjar.equals("os")) {
            System.out.println("Hmmmm, gràcies");
        } else {
            System.out.println("Ho sent, a mi dona'm ossos");
        }
    }

    // Si troba un contrincant es bregarà segons el sexe
    void bregaAmb(Gos contrincant) {
        if (this.sexe.equals("femella")) {
            System.out.println("No m'interessa bregar-me");
        } else {
            if (contrincant.diMeSexe().equals("femella")) {
                System.out.println("No em bregue amb gosses");
            } else {
                System.out.println("Vine cap ací que t'ho vaig a explicar!!!");
            }
        }
    }

}
```

PROGRAMA PRINCIPAL

```
package UF08_Exemple01;

/**
 * UF08 Exemple 01: Programa que crea gossos i descriu el seu comportament
 */
public class ProgramaGossos {
    public static void main(String[] args) {

        Gos snoopy = new Gos("mascle");
        System.out.println("- ¡Hola gos!");
        snoopy.lladra();
        System.out.println("- Tin una coca");
        snoopy.menjar("coca de segí");
        System.out.println("- Tin un os, a veure si açò t'agrada");
```

```
        snoopy.menjar("os");
        Gos scooby = new Gos("mascle");
        System.out.println("- Scooby, tin sopeta de verdures");
        scooby.menjar("sopa de verdures");
        Gos laika = new Gos("femella");
        System.out.println("- Laika,¿estàs contenta?");
        laika.cua();
        System.out.println("- A veure com lladreu");
        snoopy.lladra();
        scooby.lladra();
        laika.lladra();
        System.out.println("- A veure com us porteu entre vosaltres");
        System.out.println("Snoopy li diu a Laika: ");
        snoopy.bregaAmb(laika);
        System.out.println("Laika li diu a Scooby: ");
        laika.bregaAmb(scooby);
        System.out.println("Scooby li diu a Snoopy: ");
        scooby.bregaAmb(snoopy);
    }
}
```

- Crea un altre gos.
- Crea un mètode per a que els gossos udolen.
- Crea un mètode setter per a assignar l'edat als gossos i un altre getter per a que ens informe.
- Crea un mètode on dos gossos vagen a menjar. Si el gos que parla és major que el que escolta dirà "Primer menge jo". Si és menor dirà "Menja tu primer". Si tenen la mateixa edat fes el que consideres (pot ser que es breguen segons el sexe, etc.)

Exemple 02

Programa que gestiona contenidors de líquid i permet passar líquid de un al altre tenint en compte la capacitat lliure del segon contenidor.
La classe contenidor tindrà dos atributs: capacitat i contingut.
Cada cub es podrà crear amb una capacitat.
Tindrem mètodes setter per a plenar el contenidor(plenaContenidor), actualitzar el contingut del contenidor amb un número de litres (actualitzaContenidor) o buidar completament el contenidor (buidaContenidor) i getter que ens donarà els litres de capacitat i contingut (mostraCapacita i mostra Contingut).
Tindrem un mètode per a passar litres d'un contenidor a altre (passaContingut) de forma que si cap es passa tot i sinó es passa sols el que es puga fins que quede ple el segon.
Finalment, tindrem un mètode que dibuixe els contenidors amb el seu contingut (dibuixaContenidor). Podem utilitzar una funció que cridarem de forma repetitiva per a mostrar els contenidors.

CLASSE CONTENIDOR

```
package UF08_Exemple02;

/**
 * UF08 Exemple 02: Classe Contenidor
 */
public class Contenidor {

// Atributs
    int capacitat; // capacitat màxima en litres del contenidor
    int contingut; // contingut actual en litres


// Mètodes
    // Constructor
    Contenidor (int par_capacitat) {
        this.capacitat = par_capacitat;
    }

    // Mètodes getter
    int mostraCapacitat() {
        return this.capacitat;
    }

    int mostraContingut() {
        return this.contingut;
    }
```

```
    // Mètode setter
    void actualitzaContingut(int litres) {
        this.contingut = litres;
    }

    // Buidar el contenidor
    void buidaContenidor () {
        this.contingut = 0;
    }

    // Plenar el contenidor al màxim de capacitat
    void plenaContenidor () {
        this.contingut = this.capacitat;
    }

    // Passa els litres que pot d'un contenidor a altre
    void passaContingut (Contenidor receptor) {
        int enCaben = receptor.mostraCapacitat() - receptor.mostraContingut();
        if (enCaben > 0) {

            // Si cap tot el contingut orige en el receptor el passem i buidem el contenidor orige
            if (this.contingut <= enCaben) {
                receptor.actualitzaContingut (receptor.mostraContingut() + this.contingut);
                this.buidaContenidor();

            // Si sols cap una part, plenem el receptor complet i descomtem el que hem passat en el contenidor orige
            } else {
                receptor.plenaContenidor();
                this.contingut -= enCaben;
            }
        }
    }


    // Mètode per a dibuixar el contenidor
    void dibuixaContenidor () {

        System.out.println();
        for (int i = this.capacitat; i > 0; i--) {

            if (i>this.contingut){
                System.out.println("||      ||");
            } else {
                System.out.println("||~~~~~~||");
            }
        }
        System.out.println("==========\n");
    }
}
```

PROGRAMA PRINCIPAL

```
package UF08_Exemple02;
import java.util.Scanner;
/**
 * UF08 Exemple 02: Programa
 */
public class ProgramaContenidor {

    public static void main(String[] args) {


        // Creem dos contenidors
        Contenidor diposit1 = new Contenidor (4);
        Contenidor diposit2 = new Contenidor (7);

        // Mostrem l'estat inicial dels contenidors
        mostremContenidors (diposit1, diposit2, 1);

        // Plenem el primer contenidor
        System.out.println("Plenem el Diposit 1.");
        diposit1.plenaContenidor ();

        // Mostrem l'estat dels contenidors
        mostremContenidors (diposit1, diposit2, 2);

        // Passem el contingut del Diposit1 al Diposit2
        System.out.println("Passem el Diposit 1 al Diposit 2.");
        diposit1.passaContingut(diposit2);

        // Mostrem l'estat dels contenidors
        mostremContenidors (diposit1, diposit2, 2);

        // Tornem a plenem el primer contenidor
        System.out.println("Tornem a plenem el Diposit 1.");
        diposit1.plenaContenidor ();

        // Mostrem l'estat dels contenidors
        mostremContenidors (diposit1, diposit2, 2);



        // Tornem a passar el contingut del Diposit1 al Diposit2, però ara ja no cap tot
        System.out.println("Tornem a passar el Diposit 1 al Diposit 2.");
        diposit1.passaContingut(diposit2);

        // Mostrem l'estat dels contenidors
        mostremContenidors (diposit1, diposit2, 2);
    }
```

```
    public static void mostremContenidors (Contenidor diposit1, Contenidor diposit2, int missatge){

        Scanner intro = new Scanner (System.in);

        if (missatge==1){
            System.out.println("El Diposit 1 té una capacitat de " + diposit1.mostraCapacitat() + " litres.");
        } else {
            System.out.println("El Diposit 1 té un contingut de " + diposit1.mostraContingut() + " litres.");
        }
        diposit1.dibuixaContenidor();

        if (missatge==1){
            System.out.println("El Diposit 2 té una capacitat de " + diposit2.mostraCapacitat() + " litres.");
        } else {
            System.out.println("El Diposit 2 té un contingut de " + diposit2.mostraContingut() + " litres.");
        }
        diposit2.dibuixaContenidor();

        System.out.println ("Polsa INTRO per continuar.");
        intro.nextLine();

    }

}
```

- Canvia les capacitats.
- Crea un altre contenidor i intercanvia líquid entre els tres.

ENCAPSULAMENT

- Realitza proves sobre la visibilitat dels atributs capacitat i contingut:
    - Tal i com estan definits, es pot accedir des de una altra classe dins del mateix paquet? La resposta és sí, de fet si en el programa principal substituïm *diposit1.mostraCapacitat()* per *diposit1.capacitat* no dona cap problema.
    - I si passem el programa principal a un altre paquet, continuarem podent accedir de la forma indicada en el punt anterior? No, en aquest cas necessitarem que estiguen declarades com a **public**. Fes la prova.
    - Fes ara la prova (independent de l'anterior) i declara aquests atributs com **private**, què li passa al programa principal si intentes fer el canvi del punt primer?.

Exemple 03

Programa que permet gestionar encomandes de pizzes, així com el servici d'aquestes una vegada cuinades.
Les pizzes tenen els atributs tipus (margarida, napolitana, marinaza i calzone), tamany (mitjana i familiar) i estat (encomanada i servida).
Portarem contadors amb els totals de pizzes encomanades i servides.
Tindrem un mètode constructor de forma que al crear l'encomanda de pizza es crearà amb l'estat encomanada i s'actualitzarà el contador d'encomandes.
Tindrem un mètode per a fer el servici, si la pizza estava encomanada, passarà a estar servida i s'actualitzarà el contador de servides. Si ja estava servida es mostrarà un missatge d'avis.
Es podrà mostrar tota la informació de la pizza amb un missatge amb preformat.
Es podrà mostrar també els contadors de encomandes i servicis.

CLASSES DE TIPUS ENUMERATS

```
package UF08_Exemple03;
/**
 * UF08 Exemple Pizza: Tipus enumerat per al Tamany de pizza
 */

public enum Tamany {
   MITJANA, FAMILIAR
}
```

```
package UF08_Exemple03;
/**
 * UF08 Exemple Pizza: Tipus enumerat per al Tipus de pizza
 */

public enum Tipus {
   MARGARIDA, NAPOLITANA, MARINARA, CALZONE
}
```

```
package UF08_Exemple03;
/**
 * UF08 Exemple Pizza: Tipus enumerat per al Estat de la pizza
 */

public enum Estat {
   ENCOMANADA, SERVIDA
}
```

CLASSE PIZZA

```
package UF08_Exemple03;
/**
 * UF08 Exemple Pizza: Classe Pizza
 * Atributs: tamany (mitjana, familiar), tipus (margarida, napolitana, marinara, calzone), estat
* (encomanda, servida)
 * Variables de classe: nombre total de pizzes encomanades i servides
 */
public class Pizza {

    // Atributs de classe
    private static int totalEncomanades = 0;
    private static int totalServides = 0;

    // Atributs d'instància
    private Tamany tamany;
    private Tipus tipus;
    private Estat estat;

    // Mètode constructor
    public Pizza(Tipus tipus, Tamany tamany) {
        this.tipus = tipus;
        this.tamany = tamany;
        this.estat = Estat.ENCOMANADA;
        Pizza.totalEncomanades++;
    }

    // Mètode per a mostrar el contingut dels objectes al crear-los
    public String toString() {
        return "Encomanda de pizza del tipus " + this.tipus + ", de tamany " + this.tamany + ", i
està " + this.estat;
    }

    // Mètode per a donar l'avis de servici
    public String mostraServici() {
        return "SERVIDA: " + this;
    }
```

```
    // Mètodes getter
    public static String mostraTotalEncomanades() {
        return "El total de pizzes encomanades és: " + Pizza.totalEncomanades;
    }

    public static String mostraTotalServides() {
        return "El total de pizzes servides és: " + Pizza.totalServides;
    }

    // Altres mètodes
    public void servirPizza () {
        if (this.estat == Estat.ENCOMANADA) {
            this.estat = Estat.SERVIDA;
            Pizza.totalServides++;
        } else {
            System.out.println("Aquesta pizza ja està servida");
        }
    }

}
```

PROGRAMA PRINCIPAL

```
package UF08_Exemple03;
/**
 * UF08 Exemple Pizza: Classe Pizza
 *  Aques programa crea encomandes de pizzes (objectes), mostra les encomandes (mostra l'objecte mitjançant el mètode toString)
 *  i després les va servint. Finalment mostra l'estat actual d'encomandes i servicis.
 */
public class ProgramaPizza {

    public static void main(String[] args) {

        // Ens fan una primera encomanda
        Pizza pizza1 = new Pizza (Tipus.MARGARIDA,Tamany.MITJANA);
        System.out.println(pizza1);

        // Ens fan una segona encomanda
        Pizza pizza2 = new Pizza (Tipus.NAPOLITANA,Tamany.FAMILIAR);
        System.out.println(pizza2);
```

```
        // Ens fan una tercera encomanda
        Pizza pizza3 = new Pizza (Tipus.MARINARA,Tamany.MITJANA);
        System.out.println(pizza3);

        // Servim la segona encomanda
        pizza2.servirPizza();
        System.out.println(pizza2.mostraServici());

        // Mostrem l'estat dels nostres servicis
        System.out.println(Pizza.mostraTotalEncomanades());
        System.out.println(Pizza.mostraTotalServides());

        // Continua fent encomandes i servicis

    }

}
```