

```

//Clase main donde tengo las operaciones y llamadas principales.
import javax.swing.*;
public class main {
    public static int validarPos(Golosina[]maquina, String texto) {
//Metodo para comprobar que la posicion introducida por el usuario es
valida.
        String posicion=JOptionPane.showInputDialog(null,"Introduce
la posicion de la golosina que deseas "+texto+".\n\n"
            + "11-KitKat. 12-Chicles de fresa. 13-Lacasitos.
14-Palotes.\n"
            + "21-Kinder Bueno. 22-Bolsa Haribo. 23-Chetoos.
24-Twix.\n"
            + "31-Maiz. 32-M&M'S. 33-Papa Delta. 34-Chicles de
menta\n"
            + "41-Gusanitos. 42-Crunch. 43-Milkybar. 44-
Patatas fritas");
        for(int i=0;i<maquina.length;i++) {
            if(maquina[i].getPosicion().equals(posicion)) {
                return i;
            }
        }
        JOptionPane.showMessageDialog(null, "La posicion introducida
no es valida.");
        return -1;
    }
    public static float sumarDinero(Moneda[] cajon, Moneda[] canal) {
//Metodo para sumar todos los beneficios del cajon y el canal.
        float suma=0;
        for(int j=0;j<canal.length;j++) {
            suma=suma+((canal[j].getCantidad()-
10)*canal[j].getValor())+(cajon[j].getCantidad()*cajon[j].getValor());
        }
        suma=suma+(cajon[4].getCantidad()*cajon[4].getValor());
        return suma;
    }
    public static void main(String[]args) { //Metodo main donde defino
todos los objetos que utilizo a lo largo del programa.
        Golosina[] maquina=new Golosina[] { //Array de diferentes
golosinas que llamo maquina.
            new Golosina("KitKat",1.1f,5,"11"),
            new Golosina("Chicles de fresa",0.8f,5,"12"),
            new Golosina("Lacasitos",1.5f,5,"13"),
            new Golosina("Palotes",0.9f,5,"14"),
            new Golosina("Kinder Bueno",1.8f,5,"21"),
            new Golosina("Bolsa Haribo",1f,5,"22"),
            new Golosina("Cheetos",1.2f,5,"23"),
            new Golosina("Twix", 1f, 5,"24"),
            new Golosina("Maiz",0.7f,5,"31"),
            new Golosina("M&M'S",1.3f,5,"32"),
            new Golosina("Papa Delta",1.2f,5,"33"),
            new Golosina("Chicles de menta",0.8f,5,"34"),
            new Golosina("Gusanitos",1.5f,5,"41"),
            new Golosina("Crunch",1.1f,5,"42"),
            new Golosina("Milkybar",1.1f,5,"43"),
            new Golosina("Patatas fritas",1.1f,5,"44"),
        };
        Moneda[] canal=new Moneda[] { //Array de las diferentes
monedas que entran en los canales.

```

```

        new Moneda(1f,10),
        new Moneda(0.5f,10),
        new Moneda(0.2f,10),
        new Moneda(0.1f,10),
    };
    Moneda[] cajon=new Moneda[] { //Mismo array del canal, pero
sumandole las monedas de dos euros al cajon.
        new Moneda(2f, 0),
        new Moneda(1f,0),
        new Moneda(0.5f,0),
        new Moneda(0.2f,0),
        new Moneda(0.2f,0),
    };
    Administrador[] admin=new Administrador[] { //Array de las
diferentes contraseñas establecidas para acceder a la maquina como
administrador.
        new Administrador("1DAW3"),
        new Administrador("MaquinaPlaiaundi"),
    };
    boolean salir=false;
    int opcion;
    int i;
    String texto=""; //Texto que aparecera como comprar o mostrar
depende de la operacion que queramos realizar.

    while(!salir) { //Menu principal que aparece al ejecutar el
programa.

        opcion=Integer.parseInt(JOptionPane.showInputDialog(null,"Indica la
operacion que desea realizar:\n"
            + "1. Pedir golosina.\n"
            + "2. Mostrar golosina.\n"
            + "3. Rellenar golosina.\n"
            + "4. Apagar maquina."));
        switch(opcion) {
            case 1: //Al elegir la primera opcion del menu nos
permite comprar una golosina siempre que la cantidad de la golosina sea
mayor que 0.
                texto=" comprar";
                int pos=validarPos(maquina, texto);
                if(pos != -1) {
                    if(maquina[pos].getCantidad(>0) {

                        Pagos.pagar(maquina[pos],canal,cajon);
                    }
                    else {

                        JOptionPane.showMessageDialog(null, "No nos queda este producto,
elija otro.");
                    }
                    break;
                }
                break;
            case 2: //Al elegir la segunda opcion del menu muestra
todos los nombres de las golosinas existentes y sus cantidades siempre
que introduzcamos la contraseña 1DAW3 primero.
                String mostrar="";

```

```

        String
contraseña1=JOptionPane.showInputDialog(null, "Introduce la contraseña de
administrador.");
        if(admin[0].getContraseña().equals(contraseña1)) {
            for(i=0;i<maquina.length;i++) {

                mostrar=mostrar+maquina[i].getNombreGolosina()+" quedan
"+maquina[i].getCantidad()+" unidades.\n";
            }
            JOptionPane.showMessageDialog(null,
mostrar);
        }
        else {
            JOptionPane.showInternalMessageDialog(null,
"La contraseña introducida no es correcta.");
        }
        break;
        case 3: //Al elegir la tercera opcion del menu nos
permitira rellenar la golosina que deseemos siempre que utilicemos la
contraseña MaquinaPlaiaundi.
            String
contraseña2=JOptionPane.showInputDialog(null, "Introduce la contraseña de
administrador.");
            texto=" rellenar";
            if(admin[1].getContraseña().equals(contraseña2)) {
                pos=validarPos(maquina, texto);
                if(pos != -1) {
                    maquina[pos].setCantidad(5); //
Hago que se rellene la golosina elegida a su maxima cantidad, ya que no
tendria sentido rellenerlo a medias.
                }
            }
            else {
                JOptionPane.showInternalMessageDialog(null,
"La contraseña introducida no es correcta.");
            }
            break;
            case 4: //Al elegir la cuarta opcion del menu nos sacara
del programa despues de sacar los beneficios de la maquina.
                JOptionPane.showMessageDialog(null, "La suma total
del dia de hoy es de= "+ sumarDinero(cajon, canal)+"\u20ac\n Gracias por
utilizar nuestra maquina, que tenga un buen dia.");
                salir=true;
                break;
            default:
                JOptionPane.showMessageDialog(null, "Debes
introducir una opcion entre 1 y 4.");
            }
        }
    }
}

```

```

//Clase donde tengo las diferentes operaciones realizadas con el dinero.
import javax.swing.JOptionPane;
public class Pagos {

```

```

        public static float formato(float f){ //Metodo para redondear los
resultados float.
        return Float.parseFloat(String.format("%.1f", f).replace(',', ' ',
'.'));
    }
    public static void guardarMonedas(Moneda[] pago, Moneda[] canal,
Moneda[] cajon) { //Metodo para guardar las monedas en el canal o en el
cajon si no esta lleno.

        cajon[0].setCantidad(cajon[0].getCantidad()+pago[0].getCantidad());
        for(int i=1;i<5;i++) {
            canal[i-1].setCantidad(canal[i-
1].getCantidad()+pago[i].getCantidad());
            if(canal[i-1].getCantidad(>10) {

                cajon[i].setCantidad(cajon[i].getCantidad()+(canal[i-
1].getCantidad()-10));
                canal[i-1].setCantidad(10);
            }
        }
    }

    public static void vuelta(float cambio, Moneda[]canal) { //Metodo
para calcular el cambio dependiendo del dinero que introduce el usuario.
        Moneda[] vuelta=new Moneda[] { //Diferentes monedas
utilizadas para sacar el cambio. Las de dos euros se introducen
directamente al cajon.
            new Moneda(1f,0),
            new Moneda(0.5f,0),
            new Moneda(0.2f,0),
            new Moneda(0.2f,0),
        };
        while(cambio>0) { //Bucle que elige que moneda devolver,
siempre devolvera la mas grande posible hasta que llegue a 0. Elige solo
una posibilidad por bucle.
            if(cambio>=1) {
                cambio=cambio-1;
                vuelta[0].setCantidad(vuelta[0].getCantidad()+1);
                canal[0].setCantidad(canal[0].getCantidad()-1);
            }
            else if(cambio>=0.5f) {
                cambio=cambio-0.5f;
                vuelta[1].setCantidad(vuelta[1].getCantidad()+1);
                canal[1].setCantidad(canal[1].getCantidad()-1);
            }
            else if(cambio>=0.2f) {
                cambio=cambio-0.2f;
                vuelta[2].setCantidad(vuelta[2].getCantidad()+1);
                canal[2].setCantidad(canal[2].getCantidad()-1);
            }
            else if(cambio>=0.1f) {
                cambio=cambio-0.1f;
                vuelta[3].setCantidad(vuelta[3].getCantidad()+1);
                canal[3].setCantidad(canal[3].getCantidad()-1);
            }
            cambio=formato(cambio); //Aplico el metodo para
redondear el cambio.
        }
    }

```

```

        JOptionPane.showMessageDialog(null, "Tus vueltas son: \n"
//Imprimo las diferentes monedas que se devolveran y la cantidad de cada
una.
        +vuelta[0].getCantidad()+" monedas de 1\u20ac \n"
        +vuelta[1].getCantidad()+" monedas de 0.5\u20ac
\n"
        +vuelta[2].getCantidad()+" monedas de 0.2\u20ac
\n"
        +vuelta[3].getCantidad()+" monedas de 0.1\u20ac
\n");
    }
    public static void pagar(Golosina producto, Moneda[] canal,
Moneda[]cajon) { //Metodo para pagar las golosinas.
        int i=0;
        float restante=producto.getPrecioGolosina(); //Dinero que
falta por introducir para sacar la golosina.
        Moneda[] pago=new Moneda[] { //Array de las monedas que se
pueden utilizar para pagar las golosinas.
            new Moneda(2f, 0),
            new Moneda(1f,0),
            new Moneda(0.5f,0),
            new Moneda(0.2f,0),
            new Moneda(0.2f,0),
        };

        JOptionPane.showMessageDialog(null,producto.getPrecioGolosina()+"\u
20ac, por favor.");

        while((i<5) && (restante>0)) { //Bucle que me permite meter
maximo cinco monedas diferentes para pagar, en caso de no introducir el
importe necesario con menos de cinco monedas se cancelara la transaccion.
            String[] opciones = {"2\u20ac","1\u20ac",
"0.5\u20ac","0.2\u20ac","0.1\u20ac"}; //Menu con las diferentes opciones
de pago.
            int eleccion = JOptionPane.showOptionDialog(null, "Elige
las monedas que vas a utilizar.\n Te faltan "+ restante+"\u20ac",
"",JOptionPane.DEFAULT_OPTION,JOptionPane.INFORMATION_MESSAGE,null,opcion
es,opciones[0]);
            switch(eleccion) {
            case 0:
                restante=restante-2f;
                pago[0].setCantidad(pago[0].getCantidad()+1);
                break;
            case 1:
                restante=restante-1f;
                pago[1].setCantidad(pago[1].getCantidad()+1);
                break;
            case 2:
                restante=restante-0.5f;
                pago[2].setCantidad(pago[2].getCantidad()+1);
                break;
            case 3:
                restante=restante-0.2f;
                pago[3].setCantidad(pago[3].getCantidad()+1);
                break;
            case 4:
                restante= restante-0.1f;
                pago[4].setCantidad(pago[4].getCantidad()+1);

```

```

        break;
    }
    i++;
    restante=formato(restante);
}
if(i==5) {
    JOptionPane.showMessageDialog(null,"Maximo 5 monedas por
persona.");
}else { //Esta parte solo se ejecutara si se ha cerrado el
pago sin problemas.
    float cambio=-restante; //En caso de que lo que hemos
pagado sea superior a su precio, este valor se convertira en el cambio.
    JOptionPane.showMessageDialog(null, "Tu cambio es
"+cambio+"\u20ac");
    guardarMonedas(pago, canal, cajon); //Llamada al metodo
guardarMonedas.
    vuelta(cambio, canal); //Llamada al metodo vuelta.
    producto.setCantidad(producto.getCantidad()-1);
    //Quito una golosina de la cantidad total.
}
}
}

```

//Clase golosina donde cada golosina tiene su nombre, precio, cantidad y posicion, cada atributo tiene los metodos get y set necesarios.

```

public class Golosina {
    private String nombreGolosina;
    private float precioGolosina;
    private int cantidad;
    private String posicion;

    public String getNombreGolosina() {
        return nombreGolosina;
    }
    public float getPrecioGolosina() {
        return precioGolosina;
    }
    public int getCantidad() {
        return cantidad;
    }
    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }
    public String getPosicion() {
        return posicion;
    }

    public Golosina(String nombreGolosina, float precioGolosina, int
cantidad, String posicion) {
        this.nombreGolosina = nombreGolosina;
        this.precioGolosina = precioGolosina;
        this.cantidad = cantidad;
        this.posicion = posicion;
    }
}

```

```
}
```

```
//Clase donde defino las monedas con los atributos de valor y cantidad  
con sus metodos get y set.
```

```
public class Moneda {  
    private float valor;  
    private int cantidad;  
  
    public float getValor() {  
        return valor;  
    }  
    public int getCantidad() {  
        return cantidad;  
    }  
    public void setCantidad(int cantidad) {  
        this.cantidad = cantidad;  
    }  
  
    public Moneda(float valor, int cantidad) {  
        this.valor = valor;  
        this.cantidad = cantidad;  
    }  
}
```

```
//Clase administrador donde defino el atributo contraseña y su metodo  
get.
```

```
public class Administrador {  
    private String contraseña;  
  
    public String getContraseña() {  
        return contraseña;  
    }  
    public Administrador(String contraseña) {  
        super();  
        this.contraseña = contraseña;  
    }  
}
```