

Programación Orientada a Objetos

1º DAW - Programación

Marzo 22, 2021

Descripción del ejercicio

Se pide realizar una máquina expendedora de múltiples artículos.

Para ello se definen las clases que se muestran en la figura 1.

- **MaquinaExpendedora:**
 - `load` Realiza la carga de los articulos del paquete *datos*
 - `start` Arranca la aplicación y se encarga de mantenerla activa
 - `menu` Imprime el menú y solicita una opción
 - `verArticulos` Lista todos los artículos
 - `comprarArticulo` Solicita el numero del articulo y el dinero, si hay credito suficiente, lo elimina de la lista y el usuario usa el articulo.
 - `insertarArticulo` Solicita el tipo de articulo e inserta los datos para añadirlo en la lista de articulos. El objeto creado ha de ser del tipo seleccionado.
 - `actualizarArticulos` Elimina de la lista todos los *Alimentos* caducados
 - `comprobarArticulos` Ordena caducar a todos los *Alimentos* de la lista
- **Articulo:**
 - `toString` Devuelve la información de sus datos
- **Juguete:**
 - `usar` Imprime el mensaje: “Juego con mi [NOMBRE DEL JUGUETE]”
- **Tabaco:** La propiedad *unidades* se inicia por defecto a valor de 10.
 - `usar` Consume una unidad e imprime el mensaje:
 - * “Me acabo de fumar un cigarro, quedan [UNIDADES]”
 - * En caso de no haber unidades, el mensaje será -> “No quedan cigarros :)”
- **Comida:**
 - `caducar` Caduca con una probabilidad del 50%
 - `toString` Añade a la informacion del padre, si está caducado añadiendo “(caducado)” al final del texto.
- **Bebida:**
 - `usar` Imprime “ME tomo el [NOMBRE], me nutro [NUTRIENTES] gramos y engordo [GRASAS] gramos”
 - `caducar` Caduca con una probabilidad del 25%
 - `toString` Igual que *Comida*, incluyendo los nutrientes y grasas que aporta
- **Salado:**
 - `usar` Imprime “Me tomo el [NOMBRE] y me nutro [NUTRIENTES] gramos”
 - `toString` Añade a la informacion del padre " -> Esto nutre [NUTRIENTES] gramos"
- **Dulce:**
 - `usar` Imprime “Me tomo el [NOMBRE] y engordo [GRASAS] gramos”
 - `toString` Añade a la informacion del padre " -> Esto engorda [GRASAS] gramos"

Un *Alimento* no podrá usarse si está caducado, en su lugar imprime el mensaje “No se puede consumir, esta caducado”.

Todos las propiedades (variables) de los objetos han de ser accesibles a través de sus métodos GET/SET respectivos.

Todas las clases deberán implementar el constructor sobrecargado para inicializar sus datos. A excepción de la variable *caducado* que se establecerá a ‘falso’ por defecto.

Apartados

1. Creación de clases y jerarquía [1 punto]
2. Variables, constructores [1 punto]
3. Métodos genéricos (getter/setter y toString) [1 punto]
4. Método usar [1 punto]
5. Métodos definidos en la interfaz [1 punto]
6. Métodos de MaquinaExpendedora [5 puntos]
 - load, start y menu [1 punto]
 - Ver articulos [1 punto]
 - Comprar articulo [1 punto]
 - Introducir articulo [1 punto]
 - Actualizar y comprobar articulos [1 punto]

Menús

Menu de la maquina expendedora

1. Ver articulos
2. Comprar articulo
3. Introducir articulo
4. Poner al día
5. Caducar consumibles
0. Salir

Menu de añadir articulo

1. Juguete
2. Tabaco
3. Salado
4. Dulce
5. Bebida

Datos de prueba a importar

```
package datos;
import clases.*;
public class Articulos{
    public static final Artículo[] DATOS_ARTICULOS = {
        new Juguete("juguete", "J", 1.23),
        new Tabaco("tabaco", "T", 1.23),
        new Salado("salado", "S", 1.23, 100),
        new Dulce("dulce", "D", 1.23, 200),
        new Bebida("bebida", "B", 1.23)
    };
}
```

Diagrama

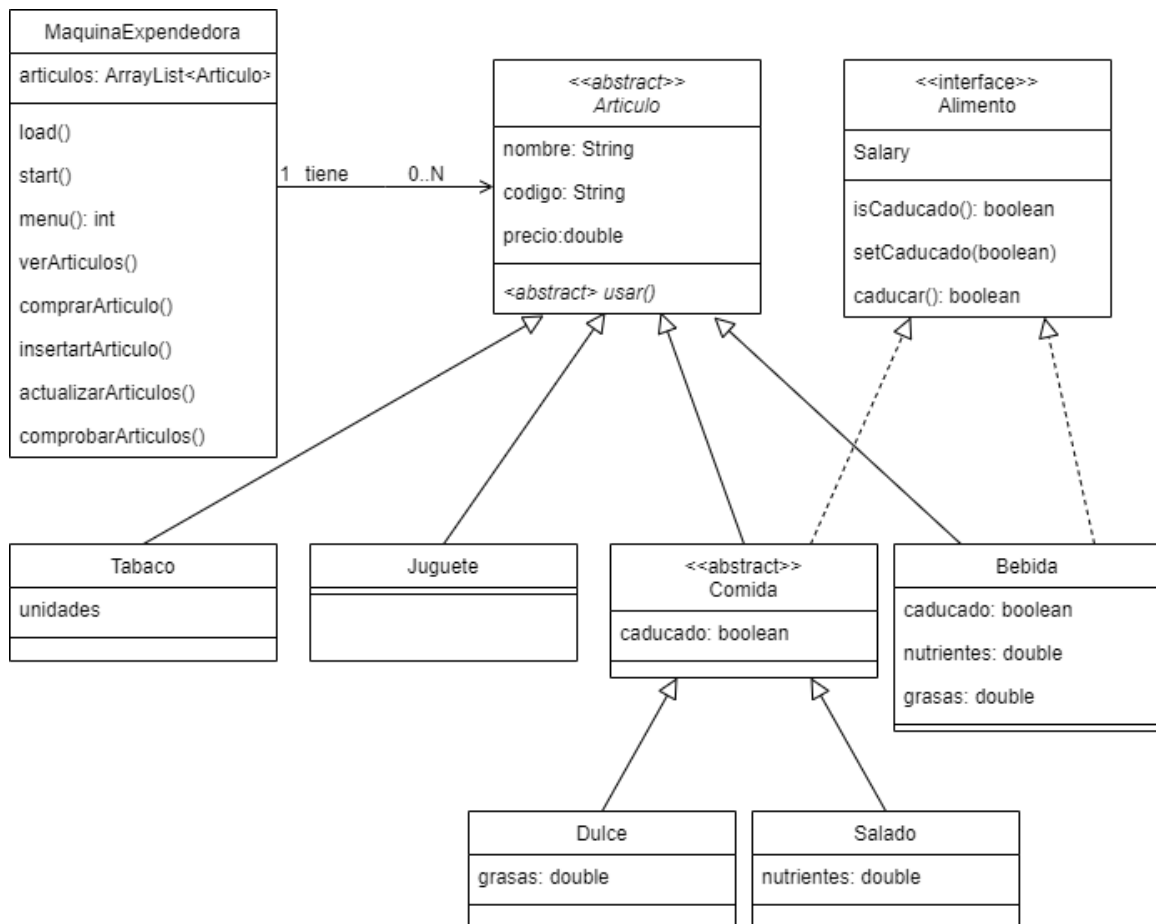


Figure 1: Diagrama de clases