

Práctica 4:

Máquina expendedora.

1. El problema

Se quiere realizar un programa para simular el funcionamiento de una máquina expendedora de botellas de agua.

2. Objetivos

- La implementación de clases de negocio (en contraposición a las clases “de librería” o “de programa” que hemos trabajado en prácticas anteriores. En este sentido se implementarán la clase Expendedora y la clase ExpendedoraMultiproducto
- Implementación de programas en los que intervienen varias clases relacionadas con relaciones “tiene – un”. La clase Expendedora y la clase Monedero se relacionan de esa forma.
- Implementación de programas en los que se encuentran separados la capa de negocio y la capa de presentación. La clase Expendedora será capaz realizar las operaciones típicas de una máquina expendedora, pero no interactuará con el usuario. El interfaz de usuario lo proporcionará una clase distinta.

3. Fases de la elaboración de la práctica.

Para realizar la práctica seguiremos los siguientes pasos:

1. Diseñar la clase Expendedora con las indicaciones que se dan más adelante.
2. Probar la clase Expendedora usando el programa TestExpendedoraGrafico que se te proporciona junto a enunciado.
3. Diseñar un interfaz para la expendedora en modo texto TestExpendedoraTexto.

(Ampliacion)

4. Diseñar la clase ExpendedoraMultiproducto
5. Incorporar la clase TestExpendedoraMultiproducto

4. Diseño de la clase Expendedora

4.1. Funcionamiento de una expendedora

La clase Expendedora soporta las operaciones típicas que se realizan sobre una máquina expendedora sencilla, que es capaz de dispensar únicamente un producto.

El funcionamiento de una expendedora, a grandes rasgos, es el siguiente:

1. El cliente introduce dinero en la máquina. Al dinero introducido lo llamaremos crédito.
2. Pulsa un botón para comprar.
3. Si hay stock del artículo seleccionado, la máquina dispensa el artículo elegido y devuelve el importe sobrante (diferencia entre el crédito introducido y el precio del artículo).

Durante el proceso se pueden producir diversas incidencias, como por ejemplo, que el cliente no haya introducido suficiente crédito para comprar el producto, que no quede producto o que no haya cambio suficiente para la devolución. La máquina también da la posibilidad de solicitar la devolución del crédito sin realizar la compra.

4.2. La clase *Expendedora*

En primer lugar, indicar que la clase *Expendedora* no realiza ninguna interacción con el usuario, no solicita ningún dato ni muestra información por pantalla.

Crear en el proyecto la clase *Expendedora* y definir los atributos y métodos que se indican a continuación.

Atributos (privados)

- crédito: Cantidad de dinero (en euros) introducida por el comprador.
- stock: Número de unidades que quedan en la máquina disponibles para la venta. Se reducirá con cada nueva venta.
- precio: Precio del único artículo que dispensa la máquina (en euros).
- cambio: Cambio del que dispone la máquina. El cambio disponible se reduce cada vez que se devuelve al cliente la diferencia entre el crédito introducido y el precio del producto comprado. **El cambio nunca se ve incrementado por las compras de los clientes:** El dinero (crédito) introducido para la compra pasa a formar parte de la recaudación y nunca se utiliza para dar cambio.
- recaudación: Representa la suma de las ventas realizadas por la máquina (en euros). Se ve incrementada con cada nueva compra.

Métodos

- Constructor: *public Expendedora (double cambio, int stock, double precio)*. Crea la expendedora inicializando los atributos cambio, stock y precio con los valores indicados en los parámetros. El crédito y la recaudación serán cero.
- Consultores: Métodos consultores para los atributos crédito, cambio, recaudación, stock y precio.
- Modificadores: Para simplificar, consideramos que los atributos de la máquina solo van a cambiar por operaciones derivadas de su funcionamiento, por lo que **no** proporcionamos modificadores públicos
- Otros métodos
 - *public String toString()* Devuelve un String de la forma:
"Credito: 3.0 euros
Cambio: 12.73 euros
Stock: 12 unidades:
Recaudación: 127.87 euros "

- *public void introducirDinero(double importe)* Representa la operación mediante la cual el cliente añade dinero (crédito) a la máquina. Esta operación incrementa el crédito introducido por el cliente en el importe indicado como parámetro.
- *public double solicitarDevolucion()* Representa la operación mediante la cual el cliente solicita la devolución del crédito introducido sin realizar la compra. El método devuelve la cantidad de dinero que se devuelve al cliente.
- *public double comprarProducto() throws NoHayCambioException, NoHayProductoException, CreditoInsuficienteException.* Representa la operación mediante la cual el cliente ordena la compra. El método devuelve la cantidad de dinero que se devuelve al cliente.

Si no se produce ninguna situación inesperada, se reduce el stock del producto, se devuelve el cambio, se pone el crédito a cero y se incrementa la recaudación.

Si la venta no es posible se lanzará la excepción correspondiente a la situación que impide completar la venta.

5. Prueba con la clase TestExpendedoraGUI

Si la clase expendedora se ha implementado bien, los programas TestExpendedoraGUI y TestExpendedoraGUIconos funcionarán correctamente. Se trata de dos programas que, a través de un interfaz gráfico de usuario (GUI) interactúan con un objeto expendedora de la clase que tú has creado.

6. Crear un interfaz para la expendedora en modo texto

En este punto, el objetivo es crear un programa interactúe con un objeto Expendedora pero, en este caso, en modo texto. El programa tendrá que ser robusto ante posibles equivocaciones cometidas por el usuario.

El funcionamiento del programa será el siguiente:

- Se creará un objeto expendedora con el cambio, stock y precio que consideres oportuno.
- Se mostrará un menú como el siguiente:
 - 1.- *Introducir dinero*
 - 2.- *Solicitar devolución*
 - 3.- *Comprar producto*
 - 4.- *Fin*

Elija una opción
- Si el usuario elije 1, se le solicitará el importe y se introducirá en la expendedora y se volverá a mostrar el menú
- Si el usuario elije 2, se solicitará el reintegro a la expendedora y se volverá a mostrar el menú
- Si el usuario elije 3, se realizará la compra del producto, comunicando las posibles incidencias expendedora y se volverá a mostrar el menú
- Si el usuario elije 4, el programa terminará.
- Si se elige una opción incorrecta, se indicará al usuario y se volverá a mostrar el menú:
- Tras cada operación se mostrará el estado de la máquina
Credito: 3.0 euros

Cambio: 12.73 euros
Stock: 12 unidades:
Recaudación: 127.87 euros

7. Diseñar la ExpendedoraMultiproducto

La clase Expendedora que se ha diseñado en la práctica solo permite la dispensación de un producto. En esta ampliación se propone diseñar una expendedora que permita dispensar N productos.

Para desarrollar esta clase habrá que seguir los siguientes pasos:

1. Crear la clase ExpendedoraMultiproducto como una **copia** de la clase Expendedora, y realizar los siguientes cambios:
 - Añadir un atributo numProductos, que representa el número de productos distintos que es capaz de dispensar la máquina.
 - Los atributos precio y stock pasarán a ser arrays de numProductos elementos.
 - getStock() y getPrecio() recibirán un parámetro que indique el número de producto que se quiere consultar (entre 1 y numProductos)
 - El método toString tendrá que mostrar el precio y el stock de todos los productos. Además mostrará también el número de productos distintos que expende la máquina.
 - El método comprar recibirá un parámetro que indique el número de producto que se quiere comprar.
 - También habrá que adaptar el constructor de la clase para poder crear la máquina con valores en los precios y stocks de cada producto.
2. Crear la clase TestExpendedoraMultiproducto como una **copia** de la clase TestExpendedoraTexto:
 - Cuando el usuario seleccione “comprar” habrá que preguntarle el número del producto que quiere comprar. Si el número del producto que indica es incorrecto, se le volverá a solicitar.