Práctica 3:
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 Monedero

- 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 Monedero
5. Incorporar la clase Monedero a la Expendedora

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 TestExpendedoraGUIIconos funcionarán correctamente. Se trata de dos programas que, a través de un interfaz grafico 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 interactué con un objeto Expendedora pero, en este caso, en modo texto.

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 elije una opción incorrecta, se indicará al usuario y se volverá a mostrar el menú:
- Para ello, se implementará un método prívate int menú() que muestra el menú al usuario, deja que introduzca un valor y devuelve el número de opción que ha elegido. Si el usuario elije una

opción incorrecta (valor no numérico o que no está entre 1 y 4) el método informará del error y volverá a mostrar las opciones hasta que el usuario introduzca una opción correcta.

- 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 clase Monedero (Ampliación)

Un monedero es un elemento que se incrusta en distintas máquinas que devuelven monedas. Dispone de un número de monedas de distinto importe. Cuando se solicita un importe al monedero, éste entrega la menor cantidad de monedas posible, es decir, intenta completar el dinero a entregar con monedas del máximo valor y, si no es posible, utiliza monedas de menor importe sucesivamente.

Diseñar la clase Monedero con los siguientes atributos y métodos

Atributos (privados)

- monedasDe100: Cantidad de monedas de 1€ (100 cent) que tiene el monedero.
- monedasDe050: Cantidad de monedas de 0.50€ (50 cent) que tiene el monedero
- monedasDe010: Cantidad de monedas de 0.10€ (10 cent) que tiene el monedero
- monedasDe001: Cantidad de monedas de 0.01€ (1 cent) que tiene el monedero

Métodos (públicos)

- public Mondero(int de100, int de050, int de 010, int de001): Crea un monedero que tiene, inicialmente la cantidad de monedas de cada tipo que se indica como parámetro
- métodos consultores de los distintos atributos.
- método public int getImporte() que devuelve el importe total del dinero que tiene el monedero. (suma del valor de las monedas que contiene).
- public String obtenerCambio(double importe) throws *NoHayCambioException*
  - *Lanzar la excepción si no es posible entregar el importe indicado con las monedas disponibles.*
  - *Si es posible entregar el cambio, reducir convenientemente la cantidad de monedas de cada tipo y devolver un String de la forma*

```
“Monedas entregadas:
de 1.00 euro: 2 monedas
de 0.50 euro: 1 monedas
de 0.10 euro: 2 monedas
de 0.01 euro: 0 monedas “
```

- public String toString(): Devuelve un String con el siguiente aspecto

```
“Monedas disponibles:
```

```
de 1.00 euro: 22 monedas
de 0.50 euro: 17 monedas
de 0.10 euro: 21 monedas
de 0.01 euro: 10 monedas “
```

8. Incorporar un monedero a la expendedora (Ampliación cont.)

- **Importante**: Para no afectar a la práctica “básica” no se harán cambios sobre la clase Expendedora, sino sobre una copia a la que llamaremos ExpendedoraConMonedero. Esta copia se realizará cuando la clase Expendedora esté completamente terminada y funcione correctamente.
- ¿Qué taréas tendrás que realizar en la clase ExpendedoraConMonedero? Entre otras …
    - Añadir un atributo privado de tipo Monedero e inicializarlo convenientemente en el constructor de ExpededoraConMonedero
    - Quitar el atributo cambio, que ya no se utilizará y realizar los cambios que esto ocasione. El cambio del que dispone la máquina será ahora el getImporte() del monedero.
    - En cuanto al constructor,
        - .... El que ahora tiene la clase creará el monedero con monedas de 10 y de 1 céntimos en su totalidad, por el importe correspondiente al “cambio” con el que se crea.
        - .... Se añadirá un nuevo constructor : *public ExpendedoraConMonedero (int stock, double precio,* int de100, int de050, int de 010, int de001*) que inicializa los atributos de la expendedora y del monedero con los valores indicados.*
    - En el toString de Expendedora, incluir el del monedero para poder ver cuántas monedas de cada tipo tiene la máquina.