Universidad de Valladolid

Departamento de Informática

Programación Orientada a Objetos

Primera entrega de Prácticas

Curso 2021/22

1. Supuesto práctico

En cada centro comercial y por la calle es muy habitual encontrar máquinas de vending que ofrecen no solo bebidas refrigerados o alimentos embolsados sino productos de distinta naturaleza como droguería, librería, electrónica etc.

Las máquinas de vending necesitan una gestión que facilite el control de los productos: existencias y precios así como la aplicación de estrategias de marketing gracias a la información sobre las ventas.

Por ello, la empresa ***VendingCo*** necesita un sistema de gestión de la red de sus máquinas vending. ***VendingCo*** mantiene y gestiona una serie de máquinas vending dispersas por la ciudad.

En la Figura adjunta se puede ver una imagen de una máquina de vending con unas características muy similares. Tal y como fue concebida esta máquina, el pago solamente se realiza con tarjeta monedero.

Figura 1 Diversas máquinas vending

1.1. Funcionalidad a proveer

Recordamos que no se está implementando la interfaz de uso de las máquinas vending, sino algunas de las clases del dominio que controlan el sistema. Se solicitan que se programen tres clases a las que llamaremos VendingSystem, VendingMachine y Product
La clase `VendingSystem` debe proporcionar la funcionalidad que permita como mínimo:

- añadir una nueva máquina vending
- eliminarla a partir de su identificador
- obtener una lista de todas las máquinas vending que gestiona
- saber el número de máquinas vending operativas
- obtener la lista de las máquinas vending que tengan alguna línea vacía.

UVa **Universidad** de **Valladolid.** Dpto. de Informática. Edificio de Tecnologías de la Información y las Telecomunicaciones, Campus Miguel Delibes. 47011 Valladolid. Tfno. 983.423000/Fax 983.423671.
e-mail:marga@infor.uva.es

La clase `VendingMachine` corresponde a la funcionalidad de la máquina vending. Cada máquina está caracterizada por un identificador y un estado (operativo o fuera de servicio). Cada máquina de vending consta de varias líneas. Cada línea tiene un identificador único y contiene varios elementos del mismo producto. Eventualmente alguna(s) línea(s) (o incluso todas) podrá(n) estar vacía(s). Es habitual que haya productos que se vendan con más frecuencia que otros, por eso, en la máquina se pueden tener varias líneas que contengan el mismo producto[1].

Un código cliente puede preguntar a la máquina, dado un identificador de línea, el precio del producto que contiene la línea así identificada.

Un código cliente puede solicitar a la máquina comprar un producto, aportando un identificador de línea y una tarjeta monedero. La máquina comprobará si hay suficiente producto en la línea correspondiente, lo sacará de la línea y descontará su precio de la tarjeta monedero. Si no hay suficiente producto, se considerará una situación no válida. Asimismo, si el saldo de la tarjeta monedero no es suficiente para abonar el precio del producto, la máquina no sacará el producto de la línea y se considerará una situación no válida.

Hay que pensar no solo en la funcionalidad de la máquina vending cuando se pide un producto, sino también en el proceso de reabastecimiento, etc.

La clase `Product` tiene asociado un precio, una fecha de consumo preferente, un nombre y un código de producto UPC (Universal Product Code)[2].

1.2. Sobre el UPC

El UPC es un identificador único para un producto. Un UPC correcto es un número de 12 dígitos. Los primeros 6 a 9 dígitos forman el código de la compañía. Los siguientes dígitos hasta el 11 se utilizan para identificar el producto dentro de los que produce dicha compañía. Por último el dígito que aparece en la posición número 12 es el dígito de control[3]. El dígito de control se calcula a partir de los 11 primeros dígitos del UPC mediante el siguiente proceso:

1. Multiplica cada dígito n_i (los primeros 11) de la siguiente forma:

$$s = n_1 \times 3 + n_2 \times 1 + n_3 \times 3 + n_4 \times 1 + n_5 \times 3 + +n_6 \times 1 + n_7 \times 3 + n_8 \times 1 + n_9 \times 3 + n_{10} \times 1 + n_{11} \times 3$$

2. Aproxima el resultado de la suma s al múltiplo de 10 más cercano. Sea m dicho múltiplo de 10.
3. Resta la suma s obtenida del múltiplo de 10 más cercano m: $d = |s - m|$.
4. El resultado d del paso anterior es el dígito de control que estará en la posición 12 del UPC.[4].

1.3. La clase `TarjetaMonedero`

Se aporta el bytecode de una clase `TarjetaMonedero` distribuida por el proveedor del sistema de tarjetas. Se dispone de la documentación de dicha clase[5] y el bytecote (.class) para incorporar al proyecto[6].

[1] Cuando se habla de producto, una instancia de producto no es el ítem sino el tipo de producto.

[2] A modo de curiosidad en http://www.upc-search.org puede consultar por UPC y obtener información del producto así identificado.

[3] En http://www.gtin.info/check-digit-calculator/ puede comprobar introduciendo 11 dígitos cuál es el dígito de control.

[4] Este proceso se encuentra explicado para diferentes tipos de códigos en: http://www.gs1.org/how-calculate-check-digit-manually.

[5] https://aulas.inf.uva.es/mod/url/view.php?id=22485

[6] https://aulas.inf.uva.es/mod/resource/view.php?id=22463

Como puede verse, la clase `TarjetaMonedero` tiene métodos para consultar el saldo actual de la tarjeta y descontar del saldo actual. La clase `TarjetaMonedero` también tiene un constructor para crear una tarjeta con su saldo inicial y un método para cargar saldo en la tarjeta. Todos los métodos que modifican el estado de la tarjeta necesitan una credencial para ser utilizados. La credencial de descontar del saldo es diferente de la de crear la tarjeta con un saldo inicial y de cargar el saldo.
Las credenciales actuales para ambas situaciones son:

saldo inicial, cargar saldo: A156Bv09_1zXo894

descontar del saldo: 6Z1y00Nm31aA-571

En esta versión de nuestra máquina de vending no se implementará ningún mecanismo para cargar saldo en la tarjeta.

Se recuerda que no se está implementando la interfaz de uso de la máquina de vending sino las clases del dominio que controlan dicha máquina.

1.4. Clases

Se espera que las clases que forman el proyecto Eclipse de la entrega contengan una clase `VendingSystem`, una clase `VendingMachine`, una clase `Product` y las clases de test `VendingSystemTest`, `VendingMachineTest` y `ProductTest`.

Las clases `VendingSystemTest`, `VendingMachineTest` y `ProductTest` deben ser clases de prueba JUnit4.

Si la solución presentada está basada en alguna otra cla- se adicional a las mencionadas, cada clase del proyecto de- be venir acompañada de su correspondiente clase de prueba implementada mediante JUnit 4 y nombrada con el mismo esquema.

2. Condiciones de entrega

- La entrega consistirá en un único archivo `.zip`.
- El archivo contendrá el proyecto Eclipse compatible con la versión Eclipse 2020-06.
- El archivo y el proyecto debe llamarse `entrega1-idAlumno1-idAlumno2`, donde `idAlumno` se refiere al identificador de la cuenta de laboratorio de cada alumno y residir en un directorio del mismo nombre.

  Ejemplo: `entrega1-javper-margar`
- El proyecto debe compilar.
- La entrega se realizará mediante la subida del archivo zip a una tarea habilitada al respecto en el aula virtual. La entrega debe hacerse una sola vez por equipo. El equipo es responsable de decidir cuál de sus integrantes es el encargado de subir la práctica.
- El límite para la entrega de la práctica se establece en las **23:55 del 13 de noviembre de 2021**
- **No se admitirán entregas que incumplan estas condiciones**.
- En caso de incumplimiento de las condiciones anteriores, se considerará la práctica como **no presentada**.
- Es necesario que el código cumpla las convenciones de código **Java**[7].

[7] http://www.oracle.com/technetwork/java/codeconvtoc-136057.html

- Es necesario documentar las clases mediante comentarios **JavaDoc**[8].
- Es deseable que cada archivo `.java` contenido en la entrega tenga en la cabecera (comentarios **JavaDoc**) el nombre de los autores (mediante la etiqueta `@author`). Para indicar el nombre, se preferirá el identificador de la cuenta de laboratorio de cada alumno en lugar de su nombre completo.

2.1. Aclaración relativa a las defensas

Las defensas deberán realizarse a lo largo de la **semana del 15-19 de Noviembre** [9].
Es necesario que todos los miembros del equipo estén presentes en la defensa.
En caso de no realizar la defensa, la práctica tendrá la consideración de **no presentada**.

3. Referencias

- https://www.bic-code.org/bic-codes/
- https://www.stocklogistic.com/codigos-contenedor-maritimo/
- https://en.wikipedia.org/wiki/ISO_6346#Check_Digit

[8] http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html
[9] Si algún equipo considera haber terminado antes del 13 de noviembre y desea realizar la defensa en la semana del 8 al 12 deberá hablarlo con su profesor de prácticas antes de la entrega y acordar con él la fecha y hora de defensa.