

Programación Orientada a Objetos
Práctica 1. Curso 2016-17

1. Supuesto práctico

Una 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 producto¹. 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.

Cada producto tiene asociado un precio, un nombre y un código de producto UPC (Universal Product Code)².



1.1. 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 control³. 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

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

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

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

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

1.2. Funcionalidad a proveer

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.

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 clase⁵ y el bytecote (.class) para incorporar al proyecto⁶.

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 MáquinaVending, una clase Producto y las clases de test MáquinaVendingTest y ProductoTest.

Las clases MáquinaVendingTest y ProductoTest deben ser clases de prueba JUnit.

2. Condiciones de entrega

- La entrega consistirá en un único archivo .tar, .tgz, .tar.gz, zip, rar.
- El archivo contendrá el proyecto ECLIPSE.
- El proyecto ECLIPSE deberá nombrarse con entrega1-idAlumno1-idAlumno2.
- El idAlumno se refiere al identificador de la cuenta de laboratorio de cada alumno.
- Ejemplo: entrega1-javper-margar

⁵<http://www.infor.uva.es/~yania/pub/poo/TarjetaMonedero/doc/>

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



- El proyecto deberá compilar.
- La entrega se realizará mediante la subida del archivo así obtenido en una tarea habilitada al respecto en el aula virtual.
- La fecha y hora límites para la entrega se establecen en la tarea del aula virtual y son: 4 de Noviembre de 2016 a las 23:55.
- **No se admitirán entregas que incumplan estas condiciones.**
- **En caso de incumplimiento** de las condiciones anteriores, se considerará la práctica como **no entregada**.
- Es recomendable que el código cumpla las convenciones de código JAVA⁷.
- Será necesario documentar las clases mediante comentarios JavaDoc⁸.
- 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.

⁷<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

⁸<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>