

consignas

Para el siguiente ejercicio, se debe realizar el diagrama UML y escribir el código Java. También desarrollar los métodos indicados por las funcionalidades y cualquier otro que fuera necesario para cumplir con las mismas, en las clases en que corresponda.

Se debe entregar el diagrama de clases UML en formato .uxf y el proyecto Java exportado de Netbeans en formato .zip

No se aceptarán archivos .pdf. Los archivos deben ser subidos al campus en el link de la actividad “segundo parcial”. No se recibirán archivos que se envíen por correo electrónico. La resolución no debe incluir técnicas no enseñadas en nuestras clases.

Una empresa que fabrica y vende máquinas expendedoras nos pide un programa para gestionar las actualizaciones de precios de los productos vendidos en sus máquinas. Este software estará instalado en cada expendedora. De cada máquina conocemos su nombre (no puede estar vacío) y los productos que vende. Sólo puede vender un máximo de 30 productos distintos. En esta lista de productos no puede haber repetidos.

De cada producto se sabe su identificador (un número entero) y su nombre (el mismo nunca puede estar vacío). Además los productos son Apreciables. Ser apreciable significa que va teniendo actualizaciones de precio. Ese precio se debe poder actualizar de distintas formas, aunque dicha actualización nunca deberá dar como resultado un precio negativo. Cada Actualización guarda un valor.

Cada producto guarda un historial con todas las actualizaciones de precio que ha sufrido. Cada una de ellas se denominan HistoricoDeActualizacion y guardan un nuevo precio. Por lo tanto el precio actual del producto es el de la última actualización que tuvo. La cantidad de actualizaciones que puede tener un producto es indefinida.

Las actualizaciones de precio de un producto pueden ser realizadas en tres formas distintas, pero es posible que en algún momento se puedan crear nuevas formas de actualización. Las tres formas que existen hasta el momento son las siguientes:

- Actualización directa: reemplaza directamente el precio con un valor determinado (por ejemplo en el momento de la asignación del precio inicial). A diferencia de las otras dos formas, ésta no necesita conocer el precio anterior porque directamente lo reemplaza con el nuevo.

Actualización por monto: modifica el precio del producto sumándole el monto de la actualización. Al realizar la actualización del precio necesita recibir el precio actual para usarlo como parte del cálculo.

- Actualización porcentual: modifica el precio del producto sumándole al precio actual el porcentaje de la actualización. Al realizar la actualización del precio necesita recibir el precio actual para usarlo como parte del cálculo.

Al momento de procesar el cálculo no debe permitirse que el nuevo valor sea menor que cero ni tampoco que la Actualización sea nula. En ese caso la operación se debe cancelar.

El programa debe resolver las siguientes funcionalidades:

- Agregar un producto. Recibe un producto para agregar a los que vende la máquina. Si el producto ya existe no lo agrega (no se permiten productos repetidos). La cantidad máxima de productos distintos que puede vender es de 30.

Mostrar por pantalla el listado histórico de actualizaciones. Muestra cada una de las actualizaciones que tuvo cada producto que vende la máquina. El mismo debe contener todos los datos de cada histórico de actualización.

Gestionar las siguientes excepciones en donde corresponda:

- que el nombre de la expendedora esté vacío.
- que el producto a agregar sea nulo o ya esté agregado o que ya se hayan agregado 30.
- que el nombre del producto esté vacío.
- que la actualización a efectuar sea nula.
- que el precio resultante de una actualización sea negativo.
- **IMPORTANTE:** La gestión de excepciones también incluye la previsión de la captura de las mismas en donde corresponda.
- Ajustar precio. Recibe el id del producto y la Actualización de precio a efectuar. Si la actualización puede efectuarse se agregará al historial de actualizaciones.

Se provee la clase PruebaMaquinas, con la creación del objeto Expendedora y la de sus productos ya codificada. También contiene los métodos necesarios para probar las clases diseñadas. Además se provee la clase HistoricoDeActualizacion completa.

