

Examen parcial – 12 de febrero de 2014

Tiempo disponible: 3 horas

Se está desarrollando una aplicación para gestionar los productos existentes en un mercado de abastos de pescado y marisco. La aplicación tiene la siguiente información acerca de cada uno de los productos ofertados: denominación del producto (no se puede repetir), cantidad (peso en Kg) de producto disponible para ser vendido a los minoristas y precio de venta por Kg de producto.

La aplicación debe manejar un catálogo de NPO productos ofertados (ni uno más ni uno menos) donde cada elemento representa la información de un producto. Veamos un ejemplo de catálogo con NPO = 8 productos donde figuran la denominación, el peso y el precio de cada producto:

Atún rojo	Besugo	Mejillón	Caballa	Lubina salvaje	Gamba	Merluza	Salmonete
80,000	80,700	12,500	355,000	410,000	150,000	475,000	200,400
15,95	15,95	3,00	3,00	5,85	9,95	12,50	6,30

De ejecución en ejecución la información sobre el pescado y el marisco disponible se mantiene en un archivo de texto `almacen.txt` organizado de manera que la información de un producto aparece en tres líneas consecutivas: en la primera la denominación, en la siguiente el peso y en la última el precio. Para el catálogo anterior la información en `almacen.txt` quedaría como ves a la derecha. Observa que el archivo finaliza con `XXX` como denominación.

```
Atún rojo
80.00
15.95
Besugo
80.70
15.95
...
Salmonete de roca
200.40
6.30
XXX
```

A través de la aplicación un pescadero debe poder realizar su compra. La cesta de la compra de un pescadero es una lista de exactamente `NP` Necesitados productos (ni uno más ni uno menos), donde cada elemento guarda la denominación del producto y la cantidad (en Kg) que se desea comprar. A modo de ejemplo, una cesta con `NP` Necesitados = 4 podría ser:

Mejillón	Besugo	Merluza	Gamba
12,500	20,000	60,000	160,000

La aplicación debe comenzar cargando en el catálogo la información que haya en el archivo `almacen.txt`, comprobando posibles errores (que no exista el archivo o que haya menos de NPO productos). Para ello debes implementar un subprograma `cargar()` que devuelva el catálogo con los datos cargados del archivo `almacen.txt` y un booleano que indique si la carga se ha podido realizar con éxito o no. Si hay más de NPO productos en el archivo, los sobrantes se ignoran. En caso de que haya errores, la aplicación finalizará. Si la carga tiene éxito se mostrará un menú con estas opciones:

1. Realizar la compra de un pescadero
2. Visualizar los productos agotados
3. Salir

Se permitirá realizar cada una de ellas tantas veces como sea necesario hasta que el usuario decida salir, momento en el que la aplicación guardará el contenido del catálogo en el archivo `almacen.txt`. Debes implementar un subprograma `guardar()` que guarde en el archivo `almacen.txt` los datos actuales del catálogo.

Si el usuario elige la opción 1 se leerá por teclado la información de una cesta de la compra y se realizará la compra, actualizando convenientemente el catálogo y mostrando en la pantalla el coste de la compra. Ten en cuenta: (a) que la cesta de la compra puede incluir productos que no existan en el catálogo, no pudiendo adquirirse; (b) que en la cesta también puede solicitarse un producto existente en el catálogo pero en una cantidad que puede ser superior a la disponible, no pudiéndose tampoco en este caso realizar la venta. Cada incidencia que se produzca, ya sea por intentar adquirir un producto inexistente, ya sea por querer adquirir más cantidad de la existente de un producto, debe notificarse indicando la denominación del producto y el problema surgido (producto no existente o cantidad excesiva). Deberás implementar, al menos, los siguientes subprogramas:

- ✓ `leerCesta()`: Devuelve una cesta de la compra leída de teclado.
- ✓ `buscarProducto()`: Dado el catálogo y la denominación de un producto, devuelve la posición donde se encuentra el producto (-1 si no existe).
- ✓ `realizarCompra()`: Dado el catálogo y una cesta de la compra, actualiza la información del catálogo convenientemente y muestra en la pantalla tanto cada incidencia que se produzca como el coste final de la compra.

Si el usuario elige la opción 2 se mostrará en la pantalla la denominación de todos los productos del catálogo que estén agotados. Para ello debes implementar un subprograma `mostrarAgotados()` que dado el catálogo muestre en la pantalla la denominación de todos los productos del catálogo que estén agotados.

Pon al principio del programa un comentario con tu nombre (y número de grupo). Entrega el archivo con el código fuente a través del Campus Virtual. Cuida el estilo.