

LABORATORIO DE PROGRAMACIÓN III (Curso 2011/2012)

PRÁCTICA 1

FECHA LÍMITE: Viernes 16 de marzo a las 20 h

Objetivos de la práctica:

1. Iniciación al entorno de Eclipse para programación en Java
2. Desarrollo de entornos gráficos de usuario usando Swing/Awt
3. Entrada/Salida en Java. Acceso a ficheros
4. Algoritmos voraces frente a algoritmos de vuelta-atrás

PARTE 1: APLICACION SUPERMERCADO SIN CAJA DE CAMBIO

Diseña una aplicación Java para ser utilizado por supermercados que venden frutas y verduras, carnes, pescados, bebidas y productos “varios” con la siguiente funcionalidad seleccionable a través de tres menús principales:

- **Configuración:** Como los productos y precios cambian a diario, el primer paso para utilizar la aplicación es, a través de este menú **Configuración**, leer desde fichero los productos disponibles y sus precios para ser utilizados en esa sesión. Concretamente, se ofrecen los siguientes submenús:
 - **Producto:** A su vez, a través de cinco submenús anidados se selecciona la información que se quiere obtener: las frutas y verduras, carnes, pescados, bebidas y productos “varios”. La información está guardada en un fichero de texto donde aparece la descripción de cada producto en una línea y en la línea siguiente su precio. Estos submenús nos deben permitir navegar por el sistema de archivos para seleccionar el fichero deseado donde se encuentra almacenada la información.
 - **Productos por defecto:** El sistema da opción a cargar los productos por defecto a partir de los ficheros existentes “frutasyverduras.txt”, “carnes.txt”, “pescados.txt”, “bebidas.txt” y “varios.txt” en vez de tener que navegar por el sistema de archivos para seleccionarlos.
 - **Operaciones caja:** Estas operaciones se implementarán en la segunda parte de la práctica.
- **Selección:** Una vez cargada la configuración, a través de este menú, se muestran unas tablas con los productos disponibles y sus precios, sobre las que se puede escribir el número de unidades que el cliente quiere comprar. Concretamente, el menú **Selección** dispone de los siguientes submenús:
 - **Frutas y verduras:** Este submenú incorpora al frame una tabla con la lista de frutas y verduras disponibles junto con sus precios, si no había sido ya incorporada con anterioridad. En caso contrario, no hace nada. En dicha tabla podemos anotar el número de unidades compradas de cada tipo de producto.
 - **Carnes:** Lo mismo para las carnes.
 - **Pescados:** Lo mismo para los pescados.
 - **Bebidas:** Lo mismo para las bebidas.
 - **Varios:** Lo mismo para los productos varios.
- **Caja:** Este menú nos permite emitir la factura con el total a pagar y efectuar el pago:
 - **Consultar caja:** Esta operación se implementará en la segunda parte de la práctica.
 - **Pago efectivo:** Esta opción abre una nueva ventana que nos muestra el precio total a pagar por los productos comprados y nos permite introducir con qué tipo de billetes y monedas queremos pagar (ver Figura). Cuando pulsamos en el botón PAGAR, se abre otra ventana que nos da el cambio óptimo (suponiendo que tenemos una cantidad infinita de dinero). El cambio óptimo se corresponde con el menor número de billetes y monedas. En la última figura, el cambio que se muestra corresponde con la parte 2 de la práctica.

Indicaciones sobre la interfaz gráfica de usuario

En primer lugar, diseñaremos la interfaz gráfica de usuario utilizando los componentes disponibles en `java.awt` y `javax.swing` como `JLabel`, `JButton`, `TextField`, etc. La aplicación principal será un marco de tipo `JFrame` con un menú asociado que muestre las diferentes opciones que se han explicado arriba.

En el menú **Configuración** utilizaremos el componente `javax.swing.JFileChooser` que nos permite navegar por el sistema de ficheros y seleccionar el fichero deseado.

Para mostrar las diferentes opciones en el menú **Selección** se utilizarán paneles de tipo `javax.swing.JTabbedPane`, a los que añadiremos una tabla de tipo `javax.swing.JTable` con las siguientes tres columnas:

- *producto*: muestra el nombre del producto leído desde el fichero seleccionado,
- *precio*: precio leído desde el fichero seleccionado y
- *cantidad*: el número de unidades consumidas (se deja en blanco cuando no se consume ninguna unidad).

Otras indicaciones:

- En principio puedes trabajar con números enteros suponiendo que un precio de 150, se corresponde con 1 euro y 50 céntimos.
- La disposición de los componentes dentro de los paneles se realizará utilizando cualquiera de los gestores estándar (`GridLayout`, `BorderLayout`, etc.).
- En el menú **Caja**, las distintas opciones abrirán nuevos marcos para poder seleccionar las monedas de pago, el total a pagar y una ventana de diálogo para visualizar el cambio (ver figuras).

Indicaciones sobre el acceso a ficheros

Para leer y escribir ficheros de texto, hay diferentes opciones en Java. Puedes utilizar las siguientes clases y sus métodos asociados:

- Para escribir desde un fichero, puedes utilizar `java.io.PrintWriter` que tiene una constructora que recibe como parámetro un `FileOutputStream` con el nombre del fichero a escribir.
- Para leer de fichero, puedes utilizar la clase `java.util.Scanner` que tiene una constructora que recibe como parámetro un `File` con el nombre del fichero a leer.

Estructura de paquetes y de clases

Esta primera parte de la aplicación, se organizará en dos paquetes **interfaz** y **supermercado**. En el paquete **interfaz** almacenaremos el código relacionado con la creación de la interfaz gráfica de usuario e implementaremos el método **main** que permite ejecutar la aplicación.

En el paquete **supermercado** almacenaremos tres clases cada una de ellas con sus constructoras, métodos de consulta y modificación de atributos y cualquier operación asociada:

- **Producto**: Cada objeto de la clase producto tiene tres atributos: su **nombre**, el **precio** y la **cantidad** seleccionada.
- **Categoría**: Cada categoría es un array de productos del mismo tipo (frutas y verduras, carnes, pescados, bebidas y varios) con su longitud asociada.
- **Tienda**: La tienda guarda la información de las cinco categorías anteriores.

| | |---| | PARTE 2: APLICACION SUPERMERCADO CON CAJA | |---|

En la segunda parte de la práctica, implementaremos la gestión de la caja de cambio. Añadiremos al proyecto un nuevo paquete **gestioncaja** para almacenar el código asociado. La caja se representa con un array de enteros de manera que cada posición representa un tipo de moneda (desde 1 céntimo hasta 50 euros) y su contenido el número de unidades de dicha moneda que tenemos actualmente en caja. Al arrancar la aplicación, la caja de cambio se inicializa con el estado de la última ejecución almacenado en el fichero "caja.txt". Al cerrar la aplicación, se almacena en dicho fichero el contenido final de la caja.

Debes añadir también las siguientes opciones al menú:

- En el menú **Configuración**, incluimos las opciones **Llenar caja** y **Vaciar caja** que, respectivamente, añaden 25 monedas de cada tipo a la caja y dejan sólo 5 monedas de cada tipo en caja. Estas operaciones se pueden ejecutar en cualquier momento de la ejecución.
- En el menú **Caja**, implementaremos las opciones:
 - **Consultar caja**: que muestra en una ventana el número de monedas de cada tipo que contiene la caja actual;
 - **Pago efectivo**: que calcula el cambio teniendo en cuenta el estado de la caja. Si no es posible realizar el cambio, se mostrará un mensaje de alerta. Debes implementar el algoritmo de **vuelta-atrás** que se explicará en clase para calcular el cambio.

Los detalles de implementación se ampliarán en clase.

Forma de Pago: 50E: 03 20E: 00 10E: 00 5E: 00 2E: 00 1E: 00 0.50c: 00 0.20c: 00 0.10c: 00 0.05c: 00 0.02c: 00 0.01c: 00

Total Compra 125.0 **PAGAR**

Figura 1: Frame para realizar el pago

i Su cambio: 50E: 0, 20E: 0, 10E: 0, 5E: 3, 2E: 5, 1E: 0, 50c 0, 20c 0, 10c 0, 5c 0, 2c 0, 1c 0

OK

Figura 2: Ventana de diálogo que visualiza el cambio