

TP4 Laboratorio II

El programa es la venta de 2 productos diferentes, contiene una jerarquía de clase básica, teniendo a la clase abstracta Producto como padre, de la cual se desprenden la clase pantalla y smartphone. tambien posee una clase sellada llamada Vendedora la cual almacena en una lista diferentes productos. A continuación se indicara los formularios q realizan las funciones.

El frmPrincipal es el formulario comienzo.

El FrmDatosSmartPhone es la venta del celular.

El FrmSmartPhone es el alta del celular.

El FrmDatosPantalla es la venta de pantalla.

El FrmPantalla es el alta de pantalla

El ultimo form Mostrar es la lectura de los archivos de texto.

Funcion:

Cuando empieza el formulario se carga en el form principal la cantidad de productos q están en la base y se hacen visibles en un DataGridView. Al entrar al formulario de celular o pantalla se cargan los distintos datos de la base de datos y estos formularios permiten realizar el alta de una venta, también permite la escritura y lectura de un archivo .txt y la serializacion de los datos a un archivo .xml. Al abrirse el formulario de pantalla o celular se lanza un evento llamado 'EjecutarHilo' el cual es de tipo 'DelegadoHilo' que devuelve void y no recibe parámetros. Dicho evento se asocia el método 'Hilo_ejecutarHilo' el cual se encarga de ejecutar el hilo secundario. El hilo secundario comprueba cada 1 segundos q la lista de ventas del programa sea igual a los elementos q están contenidos en el DataGridView, para que asi al guardar en un .txt o serializar la lista, se contengan en esta los mismos datos, sin que el usuario tenga que estar al tanto constantemente de actualizarla antes de querer realizar una operación y también realiza el Update para sincronizar los datos con la base de datos. Esto es posible mediante el método 'ComprobarLista', este comprueba si la cantidad de filas del DataGridView es distinta a la cantidad de elementos de la lista de productos. Si son iguales informa que los datos ya están sincronizados. En caso de no serlo, los agrega a la lista e informa que la lista fue actualizada. El forma del alta permite agregar un producto al DataGridView, se elije que tipo de producto es, que marca, etc. Tambien, cuando se intenta cerrar el form se pregunta al usuario si confirma la salida. La clase vendedora contiene 2 metodos de extensión los cuales son creados en al clase estatica 'MetodosExtension'. Estos métodos son los q permiten guardar y leer una clase de tipo vendedora en un archivo .txt. La interfaz 'IArchivo' tiene como métodos Guardar y Leer y es genérica, esta implementada en la clases 'Serializacion, que igual es genérica y en clase texto. Se reutiliza código y se aplica excepciones en todos los lugares necesarios.

Aplicación de funciones:

Excepciones: Se aplican a lo largo del programa y también se crea una excepción del tipo 'ProductosSinMarcaException' q es lanzada si se intenta cargar un producto sin marca.

Interfaces, archivos y tipos genéricos: Se aplica en la interfaz 'IArchivo' en las clases 'Serializacion' y 'Texto'.

Metodos de extensión: se encuentra en la clase 'MetodoExtension'.

Delegados y eventos: Se crea por fuera del formulario el delegado 'DelegadoHilo' que va retornar void y no recibira parámetros. Y dentro del formulario se crea el evento 'EjecutarHilo' el cual es de tipo DelegadoHilo, a dicho evento se le asocia un método y se lanza el mismo cuando inicia el programa y al cerrarlo(en el load y en el FormClosing).

Hilos: Se inicia mediante el evento 'EjecutarHilo' al comienzo del programa, esta se encarga de mantener la lista de productos actualizada con el DataGridView y luego hacer un Update de los datos para que se vean reflejados en la base de datos, esto lo realiza mediante el evento 'ComprobarLista'. Al cerrar el programa mediante el manejador de eventos FormClosing del frmDatos se comprueba si el hilo secundario esta vivo, de ser asi lanzo el evento 'EjecutarHilo' quien se encarga de abortar el hilo, para que el mismo no se siga ejecutando.

Base de datos: Conecta por un datatable y un DataAdapter la aplicación con una base de datos. Su implementación se encuentra en la clase BaseDeDatos.