José Mari Tomé 15/02/2019

Programación

PROG06 Tarea Evaluación 01. Realiza un programa en Java

Funcionalidad

- Las clases Cliente, CompraProducto y Factura funcionan correctamente.
- Todas las opciones del menú se ejecutan correctamente.
- El programa está bien estructurado y no hay redundancias.
- Se utilizan 4 métodos con paso de parámetros y al menos uno de ellos usa la sentencia return. Además a uno de ellos se le pasará un objeto de una de las clases que hemos creado u otro lo devolverá.

```
//Método para pedir un cliente recibe un objeto de la clase Scanner y
devuelve otro objeto de la clase Cliente
   //Recibe un parámetro de tipo Scanner y devuelve uno de tipo
cliente
   public static Cliente PedirCliente(Scanner leerTeclado ){
      System.out.print("Nombre del cliente o de la clienta: ");
      String nombre = leerTeclado.nextLine();
      System.out.print("Calle: ");
      String calle = leerTeclado.nextLine();
      System.out.print("Ciudad: ");
      String ciudad = leerTeclado.nextLine();
      System.out.print("Provincia: ");
      String provincia = leerTeclado.next();
      System.out.print("Código Postal: ");
      String codigoPostal = leerTeclado.next();
      System.out.println();

      return new Cliente(nombre, calle, ciudad, provincia,
      codigoPostal);
   }

 /*
Cuando se elija "Comprar un producto", pedirá el nombre y la cantidad.
Si el producto no está en la lista
también pedirá el precio y lo añadirá a la factura. En caso contrario,
solamente actualizará la cantidad comprada.
Obtendrá la cantidad actual y le añadirá la nueva.
//Recibe dos parámetros uno de tipo Scanner y otro de tipo factura
   */
   public static void ComprarProducto(Scanner teclado, Factura
factura) {
      System.out.print("\nNombre del producto: ");
      String nombre = teclado.next();
      System.out.print("Cantidad: ");
      int cantidad = teclado.nextInt();
    //Muestra el indice del producto si está
      int indice = factura.buscaProducto(nombre);
```

```
    //Si el indice es mayor a -1 el producto se modifica ya que existe
       if(indice >= 0) {
          factura.modificarCompra(indice, cantidad);
          factura.setNumProductos(factura.getNumProductos() + 1);
       }else{
       //Si no se crea una linea de compra nueva
          System.out.print("Precio: ");
          double precio = teclado.nextDouble();
          factura.addCompra(nombre, precio, cantidad);

       }
    }
```

- Se utilizará la sentencia try-catch para evitar que el programa falle por una excepción mientras se lee o escribe en los ficheros

```
    try{
// Si existe un fichero TXT para el comprador o la compradora
// leerá los productos y los guardará en el array hasta llenarlo
          if (archivo.exists()) {
             //si existe lo leeremos
             leerFichero = new Scanner(archivo);
             leerFichero.useLocale(Locale.US);// Notación americana,
números con punto decimal
             String nombre = "";
             double precio = 0.0;
             int cantidad = 0;

             while(leerFichero.hasNextLine()){
                nombre = leerFichero.next();
                precio = leerFichero.nextDouble();
                cantidad = leerFichero.nextInt();
                CompraProducto nuevaCompra = new CompraProducto(nombre,
precio, cantidad);
                //Añadimos el producto a la listaCopra
                listaCompra[numProductos] = nuevaCompra;
                //numProductos llevará la cuenta de productos
                numProductos++;
             }
          }
          ficheroLeido = true;
       } catch (FileNotFoundException excepcion) {
          System.out.println("Error al abrir el fichero");

       }finally {
          if(leerFichero != null) {
             leerFichero.close();
          }
       }
```

- Se utiliza la palabra reservada this, tanto para referenciar un atributo como un constructor.

```
    public Cliente(Cliente unCliente){
       this(unCliente.getNombre(), unCliente.getCalle(),
unCliente.getCiudad(), unCliente.getProvincia(),
unCliente.getcodigoPostal());
    }

    public void setNombre (String nombre) {
```

```
        this.nombre = nombre;
```

Nota: 6,5

Comentarios

- Están bien escritos, en los lugares apropiados y ayudan a entender el funcionamiento del programa. Hay código que es obvio por su nombre tanto en variables como en métodos que se ha decido no poner o estructuras básicas.

Nota: 1.25

Legibilidad

- El código está bien organizado, sigue la guía de estilo y es fácil de leer.

Nota: 1.5

Nota de la autoevaluación 9.25