PROG06 Tarea de evaluación 01. Realiza un programa en Java (40%)

Límite de entrega: viernes, 10 de febrero de 2023, 23:59
Ficheros requeridos: Producto.java, ProductoTest.java (Descargar)
Número máximo de ficheros: 3
Tipo de trabajo: Individual

PROG06 TE01 REALIZAR UN PROGRAMA EN JAVA (40%)

ENUNCIADO

Esta tarea se centra en el proceso de creación y testeo de la clase Producto. Para ello, se crearán y utilizarán 2 ficheros .java, Producto.java y ProductoTest.java. La estructura de las clases está definida en los archivos **Producto.java** y **ProductoTest.java**.

El fichero **Producto.java** contendrá la definición de la clase Producto que tendrá 3 atributos encapsulados (una cadena de caracteres para almacenar el nombre del producto, un double para el precio y un entero para cantidad) y los siguientes métodos para manejarlos:

```
Producto(String nombre, double precio, int cantidad)
Producto(String nombre, double precio)
Producto(Producto unProducto)
getNombre
getPrecio
getCantidad
getPrecioTotal
setCantidad(int cantidad)
```

El fichero **ProductoTest.java** por su parte, ejecutará el código que nos permitirá comprobar que la clase Producto funciona correctamente. Parte del código estará escrito pero habrá que completarlo para que realice lo siguiente.

1. En primer lugar creará un objeto de la clase Producto usando el primer constructor planteado y lo mostrará por pantalla. Necesitaremos definir correctamente el constructor y el método toString:

```
Naranjas                    1,60       5    8,00
```

2. A continuación, creará otros 2 objetos mediante los otros 2 constructores, actualizará su cantidad y los mostrará por pantalla. Necesitaremos definir correctamente el resto de constructores y el método setCantidad:

```
Chocolate                   2,35       0    0,00
Chocolate                   2,35       2    4,70
Naranjas                    1,60       0    0,00
Naranjas                    1,60      10   16,0
```

3. Después, mostrará diferentes valores del segundo producto creado. Necesitaremos definir todos los métodos get.

```
NOMBRE: Chocolate
PRECIO: 2.35
CANTIDAD: 2
PRECIO TOTAL: 4.7
```

Una vez que ya tenemos la clase Producto definida y testada, realizaremos una serie de pruebas del manejo de clases.

4. Pediremos los datos de un producto por teclado, crearemos un nuevo objeto de la clase Producto y lo mostraremos por consola. Para ello, definiremos el método pedirProducto.

```
Nombre: Pan
Precio: 1,25
Cantidad: 2
Pan                              1,25       2    2,50
```

5. A continuación, usaremos el método pedirProducto para crear un array de 3 objetos de la clase Producto y lo mostraremos por consola mediante Arrays.toString.

```
Nombre: Aceite
Precio: 3,88
Cantidad: 1
Nombre: Vinagre
Precio: 0,95
Cantidad: 1
Nombre: Sal
Precio: 1,5
Cantidad: 1
[Aceite                           3,88       1    3,88, Vinagre                          0,95       1    0,95,
Sal                              1,50       1    1,50]
```

6. También mostraremos, datos concretos de objetos concretos de ese array. En este caso, los resultados se mostrarán sin redondear:

```
Precio del elemento 1: 0.95
Nombre del elemento 0: Aceite
Cantidad del elemento 2: 1
Precio total del elemento 1: 0.95
```

7. y seguidamente el precio total de cada uno de ellos y la suma de todos ellos. En este caso, los resultados se mostrarán formateados a 2 decimales. Para ello, definiremos el método precioTotal.

```
Precio total del producto 1: 3.88
Precio total del producto 2: 0.95
Precio total del producto 3: 1.50
Suma total: 6.33
```

En este caso, la estructura de los ficheros está definida y comentada por tanto, solo se calificará la **funcionalidad** y la **legibilidad**. Recordad añadir la ***cabecera del programa*** en la clase principal, tal y como venimos haciendo previamente.

RECURSOS

Para realizar esta tarea puedes utilizar jGRASP o el editor VPL de Moodle que además te dirá si la salida del programa coincide con la propuesta.

INDICACIONES DE DESARROLLO

Realiza el programa por partes. Escribe el código poco a poco compilándolo y testeándolo frecuentemente. En este caso además, trabaja en paralelo con el fichero de la clase y el fichero de test en el que se utiliza esa clase. De está, manera podrás comprobar que los cambios realizados en la clase funcionan bien en el programa cliente.

Para definir el **método toString**, te puede venir bien el método estático format de la clase String que funciona de manera análoga a printf. Por ejemplo:

> String.format("Precio: %8.2f", precio);
>
> Genera un String con 8 caracteres. El valor contenido en la variable precio la formateará con 2 decimales y ocupará las posiciones más a la derecha. El resto se completarán con espacios en blanco.
>
> 123.4567 se mostrará como 123.45 con 2 espacios en blanco delante.
>
> Si añadimos un signo - delante del 8 el valor se alineará a la izquierda.

En los **métodos**, los objetos también se pueden pasar como parámetro o devolver con un return. Tendremos que definir correctamente el método y recoger el objeto al hacer la llamada.

Guía de estilo: (Referencia)

> Da nombres significativos a métodos, variables y parámetros siguiendo además las reglas de Java.

Utiliza correctamente la indentación y los espacios en blanco.
Incluye cabeceras explicativas al principio del programa y al principio de cada uno de los métodos, así como junto a cualquier línea de código que lo pueda necesitar.
En el caso de los métodos, como comentario, además de lo que hace indica los parámetros que utiliza y el valor que devuelve si es que existe.

INDICACIONES DE ENTREGA

El programa se escribirá directamente en el IDE proporcionado o se copiará en él una vez finalizado.

En la cabecera, además de la información que describe el programa, se añadirá el enlace a la evaluación realizada, un enlace a Google Drive o a Youtube según el formato elegido.

CRITERIOS DE CALIFICACIÓN

La tarea se evaluará siguiendo la siguiente rúbrica:

Funcionalidad (80%).

MUY BUENO (10PT)	La clase Producto funciona correctamente. ProductoTest.java: cumple con los 3 primeros apartados del enunciado. ProductoTest.java: cumple con los apartados 4 y 5 del enunciado. ProductoTest.java: cumple con los apartados 6 y 7 del enunciado. Se pasan todos los test de la plataforma.
BUENO (7-9PT)	La clase Producto funciona correctamente. ProductoTest.java: cumple con los 3 primeros apartados del enunciado. ProductoTest.java: cumple con los apartados 4 y 5 del enunciado. ProductoTest.java: cumple con los apartados 6.
SATISFACTORIO (5-7PT)	La clase Producto funciona correctamente. ProductoTest.java: cumple con los 3 primeros apartados del enunciado. ProductoTest.java: cumple con los apartados 4 y 5 del enunciado.
DEBE MEJORAR (<4PT)	No cumple las especificaciones básicas

Legibilidad (20%).

Con esta rubrica se calculará la nota de vuestro programa y se comparará con la que habéis indicado en la autoevaluación.

Si la diferencia es menor de un punto, se calculará la nota media entre las dos.
Si la diferencia es mayor, se utilizará directamente la calificación de la profesora.

En los casos en los que falte la autoevaluación, se utilizará la calificación de la profesora con una penalización de un punto.

Ficheros requeridos

Producto.java

```java
// Clase Producto.

// Atributos
   private String nombre;
   private double precio;
   private int cantidad;

// Constructor
   Producto(String nombre, double precio, int cantidad)
   Producto(String nombre, double precio)    // Supondrá cantidad = 0
   Producto(Producto unProducto)             // Supondrá cantidad = 0

// Métodos
   getNombre
   getCantidad
   setCantidad
   getPrecio
   getPrecioTotal       // Calcula y devuelve el precio total de la compra

   toString             // Formatea la compra de un producto de la siguiente manera
                        // Nombre: hueco de 30 caracteres, alineado a la izquierda
                        // Precio: hueco de 8 caracteres, precio alineado a la derecha, 2 decimales
                        // Cantidad: hueco de 8 caracteres, alineado a la derecha
                        // Precio total: hueco de 8 caracteres, precio alineado a la derecha, 2 decimales
                        // Naranjas                          1,40       5    7,00
                        // Te puede ayudar usar el método format de la clase String
```

ProductoTest.java

```java
/*
   Este programa testea la clase Producto
*/
import java.util.*;
import java.io.*;

public class ProductoTest {

   // Método principal
   public static void main(String[] args) {

      // Creamos un producto con el primer constructor
      Producto productoUno = new Producto("Naranjas", 1.6, 5);

      // Lo mostramos formateado con el método toString
      System.out.println(productoUno.toString());


      // Creamos un producto con el segundo constructor y lo mostramos
      // Producto: "Chocolate", Precio: 2.35


      // Actualizamos la cantidad del producto anterior a 2 y lo mostramos


      // Creamos un producto con el tercer constructor a partir del producto 1 y lo mostramos


      // Actualiza la cantidad del producto anterior a 10 y lo mostramos



      // Mostramos la siguiente información del producto 2
      // Nombre
      // Precio
      // Cantidad
      // Precio total
      String nombreProducto = productoDos.getNombre();
      System.out.println("NOMBRE: " + nombreProducto);


      // Pedimos datos por teclado, creamos un producto y lo mostramos
      Scanner leerDatos = new Scanner(System.in);
      Producto otroProducto = pedirProducto(leerDatos);
      System.out.println(otroProducto);

      // Creamos un array de 3 Productos, pedimos los datos y los mostramos
      Producto[] arrayProductos = new Producto[3];
      for (int i = 0; i < arrayProductos.length; i++) {
         arrayProductos[i] = pedirProducto(leerDatos);
      }
      System.out.println(Arrays.toString(arrayProductos));


      // Mostramos el precio del elemento 1 de arrayProductos
      System.out.println("Precio del elemento 1: " + arrayProductos[1].getPrecio());

      // Mostramos el nombre del elemento 0 de arrayProductos

      // Mostramos la cantidad del elemento 2 de arrayProductos

      // Mostramos el precio total del elemento 1 de arrayProductos


      // Muestra la suma del precio total de todos los productos del arrayProductos
      double suma = precioTotal(arrayProductos);
      System.out.println(suma);

   }

   // Pedimos datos por teclado y creamos un producto
   public static Producto pedirProducto(Scanner leerDatos) {


   }

   // Mostramos el precio total de cada uno de los productos del arrayde Productos
   // y devuelvemos la suma de todos ellos
   public static double precioTotal(Producto[] arrayProductos) {


   }
}
```

VPL

Navega por la unidad

◄ PROG06 Tarea de aprendizaje 02. Trabajando con arrays de objetos.

Ir a...

PROG06 Tarea de aprendizaje 03. Más sobre clases. ►

Contacta con nosotros:

Dirección: Paseo de Ubarburu 39, Edificio EnerTIC of. 206 · Donostia San Sebastián
Telefono : 945 567 953
E-mail: info@birt.eus
Twitter: @Birt_LH