

Tarea 07

Ejercicio 1:

Primero le pido por teclado al usuario un número de listas a generar, y luego un nombre para cada lista con un bucle que se repite tantas veces como listas allá.

Luego con un método le pido al usuario que pase por teclado un número entre el 0 y el 10 en bucle, y estos números se irán añadiendo a la lista, cuando se escriba Hacer media, el bucle se detendrá y se cerrará la lista. Este método se repetirá una vez por cada lista con la diferencia que después de la primera vez ya no hará falta escribir nada en concreto para hacer la siguiente porque tomará de referencia la extensión de la primera lista y pedirá en bucle valores hasta que haya rellenado todos los huecos. Tras esto llamo a otro método donde se hace la media de todos los valores de cada lista y se muestra por teclado los valores de cada lista y la media resultante.

Ejercicio 2:

Empiezo el main con la lista llamada Producto con un nombre y su precio en euros para cada artículo. Por lo que las variables y propiedades de cada objeto de la lista son el nombre, que es un string y el precio que es un double. Y con clases públicas dependiendo de si el método necesita acceder a estos datos.

Llamo al método mostrarProductos que muestra en la consola la lista con los productos que hay y su precio. Creo un array llamado ProductoCarrito que llama a un método, llenarCarrito. Lo que hace es pedirle información al usuario mediante JOptionPane. Siendo un do-while que pide el nombre de un producto y luego una cantidad de este. Y esto se repite en bucle hasta que el usuario escriba Pagar, y el bucle se cierra.

Luego llamo al método Pago, que empieza pidiendo por teclado si quiere aplicar 21 o 4% de IVA. Una vez escrita la opción pasa a un switch que dependiendo del IVA se hace una operación u otra, siendo la suma del precio de todos los productos y aplicándole el IVA escogido. Llegado a este punto se han ido añadiendo cualidades como la cantidad de un producto, el precio final de la compra y el IVA aplicado.

Tras mostrar el precio de la compra se llama al método cambioPago, el cual empieza pidiéndole al usuario la cantidad de dinero que quiere pagar, si la cantidad es inferior al precio de la compra se vuelve a pedir. Una vez escrita una cantidad válida se realiza la resta entre la cantidad ingresada y el precio de la compra para sacar el cambio. De aquí guardamos la cantidad ingresada y el cambio a devolver. Y para acabar mostramos en la consola todos los datos que he ido guardando: el IVA elegido, el precio total bruto de la compra y el precio con el IVA aplicado, la cantidad de artículos comprados, la cantidad pagada por el usuario y el cambio a devolver.

Ejercicio 3:

Creo una lista de objetos con 10 productos, con los atributos de nombre, cantidad y precio. Creo los getters y setters correspondientes y llamo a un método que muestra en la consola la lista de los productos. Luego llamo al método `añadirProducto`, que pide el nombre de un producto nuevo por teclado mediante scanner, o escribir `Entregar` para acabar. Luego pide un número entero que será la cantidad y luego un `double` que será el precio y se añade un nuevo objeto a la lista. Esto se hace en bucle con `do while` y la condición del `while` para cerrar el bucle es escribir `Entregar` como he comentado antes. Una vez cerrado el método se vuelve a mostrar la lista con los productos añadidos en la consola.

Ejercicio 4:

Creamos 2 listas `hashmap`, uno para el precio de los productos y otro para la cantidad que hay de estos. Llamamos a los 2 primeros métodos que inician la lista y la muestran en la consola. Luego con un bucle `while` mostramos en consola las 4 posibilidades que hay según el número que escribas del 1 al 4, siendo comprar, administrar stock, pagar o salir. Luego pasamos a un `switch` y cada `case` es un número y depende de cual se haya escrito se ejecutarán métodos concretos. En el primer `case` es `comprarProducto`, por scanner pedimos el nombre de un producto, pero si no hay stock te saltará un mensaje diciéndote que el producto no está disponible. Si le das al 2 se llama al método `administrarStock`, que pide el nombre de un producto, luego la cantidad que quieres añadir y la lista original se actualiza en ambos casos, tanto si quitas como si añades productos. El 3 es el método pagar que te pide el IVA que quieres aplicar, si pasamos a un `switch` que hace la operación para obtener el precio de la compra aplicando el IVA. Luego pide la cantidad de dinero con la que se va a pagar, si no es suficiente te vuelve a pedir la cantidad. Una vez se ingresa un valor adecuado se realiza la resta entre la cantidad escrita y el valor de la compra para obtener el cambio. Por último si escribes 4 el código se queda en `false` con el boolean y el código deja de ejecutarse porque está hecho de forma que se ejecuta en bucle a menos que salgas.