

	<p align="center">IES JOAN RAMIS I RAMIS Departament d'Informàtica</p>	<p align="center">EXAMEN UNITATS 9 i 10</p>
<p align="center">CFGS DAW 1r</p>	<p align="center">Programació</p>	<p align="center">7 maig 2024</p>

Donats els dos fitxers *compra1.csv* i *compra2.csv* on hi ha emmagatzemades dues compres fetes a un supermercat en forma d'una sèrie de productes, quantitats i preus unitaris, heu de fer una lectura dels fitxers i convertir-los en objectes de tipus *LiniaDeFactura* i després adjuntar-los a un altre objecte de tipus *Factura*.

Cada línia dels fitxers té el format "Descripció;Quantitat;PreuUnitari" per exemple

"Patates;5;6.25"

Les dues classes demanades han de tenir aquests atributs:

- *LiniaDeFactura*:
 - *descripcio* (String)
 - *quantitat* (double)
 - *preuUnitari* (double)
- *Factura*:
 - *data* (Date)
 - *total* (double)
 - *linies* (ArrayList Linia de factura)

Heu d'anar agafant totes les línies del fitxer i **transformar-les en objectes tipus *LiniaDeFactura***. Quan estiguin totes llegides i emmagatzemades, heu de generar seguidament un objecte de tipus *Factura* donant valor a tots els seus atributs: la **data** serà el dia d'avui (ho podeu fer creant un objecte de tipus *Date()*), el **total** l'heu de calcular operant amb les quantitats i preu unitari de cada línia i les **línies** serà un *ArrayList* amb les línies llegides del fitxer transformades en objecte de tipus *LiniaDeFactura*

Cada factura s'ha d'emmagatzemar dins un objecte de tipus *ArrayList<Factura>*.

Després d'haver llegit els dos fitxers, l'*ArrayList* que contengui les dues factures s'ha d'emmagatzemar dins d'un fitxer anomenat ***factures.dat***.

Al final, el programa haurà de **llegir les factures del fitxer *factures.dat*** i mostrar la informació que contenguin per la consola en amb el següent format.

Data: XX-XX-XXXX

Total: XXX,XX €

Línies de factura:

- xxx xxx xxx

- ...

	IES JOAN RAMIS I RAMIS Departament d'Informàtica	EXAMEN UNITATS 9 i 10
CFGS DAW 1r	Programació	7 maig 2024

Les dates es poden formatejar emprant un objecte de tipus

- `new SimpleDateFormat("dd-MM-yyyy").format(date)`

I els preus es poden formatejar emprant un objecte de tipus

- `new DecimalFormat("#.00").format(preu)`

Rúbrica:

Apartat	Malament (0-1)	Bé (2-3)	Notable (4)	Excel·lent (5)
Lectura i transformació de fitxers	No es llegeixen els fitxers o hi ha errors en la lectura.	S'han llegit els fitxers i s'han creat objectes <code>LiniaDeFactura</code> , però hi ha errors menors.	S'han llegit els fitxers i s'han creat correctament els objectes <code>LiniaDeFactura</code> .	Lectura perfecta sense errors i transformació precisa i completa de les dades en objectes <code>LiniaDeFactura</code> .
Creació d'objectes <i>Factura</i>	No s'ha creat cap objecte <code>Factura</code> o hi ha errors en la creació.	S'han creat objectes <code>Factura</code> , però hi ha errors en el càlcul del total o en altres atributs.	S'ha creat correctament l'objecte <code>Factura</code> amb tots els atributs correctes.	Creació perfecta de l'objecte <code>Factura</code> amb càlcul precís del total i altres atributs sense errors.
Emmagatzematge de les factures	No s'ha emmagatzemat les factures o hi ha errors en l'emmagatzematge.	S'han emmagatzemat les factures, però hi ha problemes amb el format o amb el fitxer.	Les factures s'han emmagatzemat correctament dins del fitxer <code>factures.dat</code> .	Emmagatzematge precís i sense errors de les factures dins del fitxer <code>factures.dat</code> .
Format de sortida	La informació de les factures no es mostra o no segueix el format correcte.	La informació es mostra però amb problemes de format o falta d'alguna part.	La informació de les factures es mostra correctament amb el format especificat.	La informació de les factures es mostra amb el format correcte, incloent dates i preus formatejats correctament.
Funcionalitat general	Hi ha errors greus en l'execució del programa o no compleix les especificacions.	El programa funciona, però hi ha errors menors o no s'executen totes les funcionalitats.	El programa compleix totes les especificacions i funciona correctament.	El programa supera les especificacions, és robust, eficient i lògicament organitzat.
Eficàcia del codi i estructura	La estructura del codi és confusa i ineficient.	El codi és estructurat, però hi ha mancances en la claredat o en l'optimització.	El codi està ben estructurat i és fàcil de llegir i comprendre.	El codi és elegant, eficient i segueix els principis de disseny de programació.

	<p align="center">IES JOAN RAMIS I RAMIS Departament d'Informàtica</p>	<p align="center">EXAMEN UNITATS 9 i 10</p>
CFGFS DAW 1r	Programació	7 maig 2024

Dados los dos archivos *compra1.csv* y *compra2.csv* donde se encuentran almacenadas dos compras realizadas en un supermercado en forma de una serie de productos, cantidades y precios unitarios, debes hacer una lectura de los archivos y convertirlos en objetos de tipo *LineaDeFactura* y luego añadirlos a otro objeto de tipo *Factura*.

Cada línea de los archivos tiene el formato "Descripción;Cantidad;PrecioUnitario" por ejemplo "Patatas;5;6.25"

Las dos clases solicitadas deben tener estos atributos:

- *LiniaDeFactura*:
 - *descripcion* (String)
 - *cantidad* (double)
 - *precioUnitario* (double)
- *Factura*:
 - *fecha* (Date)
 - *total* (double)
 - *lineas* (ArrayList Linea de factura)

Debes ir cogiendo todas las líneas del archivo y **transformarlas en objetos de tipo *LineaDeFactura***. Cuando estén todas leídas y almacenadas, debes generar seguidamente un objeto de tipo *Factura* dando valor a todos sus atributos: la **fecha** será el día de hoy (puedes hacerlo creando un objeto de tipo *Date()*), el **total** lo debes calcular operando con las cantidades y precios unitarios de cada línea y las **líneas** serán un *ArrayList* con las líneas leídas del archivo transformadas en objeto de tipo *LineaDeFactura*.

Cada factura debe almacenarse dentro de un objeto de tipo *ArrayList<Factura>*.

Después de haber leído los dos archivos, el *ArrayList* que contenga las dos facturas se debe **almacenar dentro de un archivo llamado *facturas.dat***.

Al final, el programa deberá **leer las facturas del archivo *facturas.dat*** y mostrar la información que contengan por la consola con el siguiente formato:

Fecha: XX-XX-XXXX

Total: XXX,XX €

Lineas de factura:

- XXX XXX XXX

- ...

	IES JOAN RAMIS I RAMIS Departament d'Informàtica	EXAMEN UNITATS 9 i 10
CFGS DAW 1r	Programació	7 maig 2024

Las fechas se pueden formatear utilizando un objeto de tipo

- `new SimpleDateFormat("dd-MM-yyyy").format(fecha)`

Y los precios se pueden formatear utilizando un objeto de tipo

- `new DecimalFormat("#.00").format(precio)`

Rúbrica:

Apartado	Mal (0-1)	Bien (2-3)	Notable (4)	Excelente (5)
Lectura y transformación de archivos	No se leen los archivos o hay errores en la lectura.	Se han leído los archivos y se han creado objetos LineaDeFactura, pero hay errores menores.	Se han leído los archivos y se han creado correctamente los objetos LineaDeFactura.	Lectura perfecta sin errores y transformación precisa y completa de los datos en objetos LineaDeFactura.
Creación de objetos Factura	No se ha creado ningún objeto Factura o hay errores en la creación.	Se han creado objetos Factura, pero hay errores en el cálculo del total u otros atributos.	Se ha creado correctamente el objeto Factura con todos los atributos correctos.	Creación perfecta del objeto Factura con cálculo preciso del total y otros atributos sin errores.
Almacenamiento de las facturas	No se han almacenado las facturas o hay errores en el almacenamiento.	Se han almacenado las facturas, pero hay problemas con el formato o con el archivo.	Las facturas se han almacenado correctamente dentro del archivo facturas.dat.	Almacenamiento preciso y sin errores de las facturas dentro del archivo facturas.dat.
Formato de salida	La información de las facturas no se muestra o no sigue el formato correcto.	La información se muestra, pero con problemas de formato o falta alguna parte.	La información de las facturas se muestra correctamente con el formato especificado.	La información de las facturas se muestra con el formato correcto, incluyendo fechas y precios formateados correctamente.
Funcionalidad general	Hay errores graves en la ejecución del programa o no cumple las especificaciones.	El programa funciona, pero hay errores menores o no se ejecutan todas las funcionalidades.	El programa cumple todas las especificaciones y funciona correctamente.	El programa supera las especificaciones, es robusto, eficiente y lógicamente organizado.
Eficacia del código y estructura	La estructura del código es confusa e ineficiente.	El código está estructurado, pero hay carencias en la claridad o en la optimización.	El código está bien estructurado y es fácil de leer y comprender.	El código es elegante, eficiente y sigue los principios de diseño de programación.