



La ejercitación está modelada para que sea resuelta por módulos. Estos se encuentran ordenados para ser resueltos de manera incremental, y los mismos son dependientes, hay que resolverlos en orden.

Destacado:

- **Para obtener un DESTACADO, debe resolver correctamente alguna de las consignas opcionales explicadas a continuación.**
- **Las consignas de estos enunciados, en general, requieren una investigación adicional por parte de los alumnos.**
- **Solo se puede obtener un destacado en la primer entrega. No en sucesivas reentregas.**

ACLARACIÓN:

Como se está trabajando con listas, eliminando y agregando objetos, deberás implementar el equals a las clases que sean necesarias. Además, si querés podés implementar el toString() de estos objetos.

Opción 1:

- Descargar el archivo **listadoDealumnos.csv**
- Armar una clase en Java, denominada **LectorDeArchivosCSV** que contenga un método que lea el archivo anterior y deje a los alumnos en una List<Alumno>.
- Utilizando el **DH Manager** programado anteriormente y la lista de alumnos leída desde el archivo, inscribir a los alumnos de dicha lista en el curso Mobile Android.

Aclaraciones:

1. Hay varias opciones para leer el archivo el archivo **.csv** se puede utilizar la librería **openCsv** en combinación con la librería **commons-lang** o el objeto **BufferReader**. A continuación hay referencias, con ejemplos:
 - a. Para realizar la lectura utilizando openCsv:
 - i. <https://sourceforge.net/projects/opencsv/>
 - ii. <http://howtodoinjava.com/apache-commons/parse-read-write-csv-files-opencsv-tutorial/>
 - iii. <http://www.javainterviewpoint.com/how-to-readparsewrite-csv-file-using-opencsv/>
 - iv. <https://mvnrepository.com/artifact/org.apache.commons/commons-lang3/3.4>
 - b. Para realizar la lectura utilizando BufferReader
 - i. http://chuwiki.chuidiang.org/index.php?title=Leer_fichero_CSV_con_Java
2. El curso Mobile Android crearlo directamente utilizando Java, es decir, no es necesario leerlo desde el archivo csv.
3. Una forma de importar el archivo **.jar** es siguiendo los siguientes pasos:
 - a. En el menú superior de IntelliJ ir a File > Project Structure.
 - b. Seleccionar la opción Librerías.
 - c. Hacer click en el botón + verde, situado en la parte superior de la ventana.
 - d. En las opciones del menú desplegable elegir la opción Java.
 - e. Buscar el archivo jar descargado.
 - f. Hacer click en aplicar y dar ok.

ACLARACIÓN:

Como se está trabajando con listas, eliminando y agregando objetos, deberás implementar el equals a las clases que sean necesarias. Además, si querés puedes implementar el toString() de estos objetos.

Opción 2:

Desde el departamento de Alumnos de Digital House nos pidieron informar sobre aquellos alumnos de los cuales tengamos referencias, para saber si hay que darles seguimiento. Por lo tanto agregamos al sistema una funcionalidad para identificar si determinados alumnos son estudiosos o no.

- Descargar el archivo **DigitalException.jar**
- Importar el archivo jar al proyecto
- Hacer que **Alumno** implemente la interfaz **Estudioso** que ya está definida en el jar. El método deberá informar si el alumno es estudioso o no según corresponda. (A criterio del alumno)
- En el **main** usar la función `DigitalException.inscripcionDH(estudioso);`
- Notar que necesita pasar un objeto Estudioso por parámetro.
- En caso de suceder una excepción de tipo `DHException` se deberá imprimir por pantalla el mensaje "Venir al colearning."
- En caso de suceder una excepción de tipo `OtherException` se deberá imprimir por pantalla el mensaje "Hablar con alumnos."

Aclaraciones:

1. Para importar el archivo jar se puede seguir los siguientes pasos:
 - a. En el menú superior de IntelliJ ir a File > Project Structure.
 - b. Seleccionar la opción Librerías.
 - c. Hacer click en el botón + verde, situado en la parte superior de la ventana.
 - d. En las opciones del menú desplegable elegir la opción Java.
 - e. Buscar el archivo jar descargado.
 - f. Hacer click en aplicar y dar ok.

ACLARACIÓN:

Como se está trabajando con listas, eliminando y agregando objetos, deberás implementar el equals a las clases que sean necesarias. Además, si querés puedes implementar el toString() de estos objetos.

Opción 3:

Debido a la necesidad de distribuir el material teórico y práctico de las cursadas, se decidió digitalizar el mismo y facilitar a los alumnos la impresión gracias a la inauguración de la fotocopidora de alumnos.

El material a ser digitalizado son guías prácticas y teóricas. Las mismas contienen la siguiente información:

- Guías Teóricas: código, cantidad de páginas, costo, título, lista de los temas.
- Guías Prácticas: código, cantidad de páginas, costo, título, cantidad de ejercicios.

En cada **Curso de Digital House**, ahora también se conocen los **códigos** de las guías prácticas y teóricas que el curso va a utilizar a lo largo de la cursada. Con esto logramos que un alumno ahora pueda acercarse a la fotocopidora y simplemente puede imprimirlo en la fotocopidora entregando el código de la guía. De esta forma logramos que el alumno pueda tener encima las guías impresas. La fotocopidora tiene dentro de sus propiedades, todas las guías teóricas y prácticas de todos los cursos.

Como DH está en constante crecimiento, la cantidad de guías tanto prácticas como teóricas van a ir en aumento. Esta cantidad varía muy seguido (se dan de baja guías viejas, se dan de alta guías nuevas).

Como la fotocopidora tiene escasos recursos de procesamiento, queremos implementar un sistema eficiente en donde se pueda acceder rápidamente a la guía e incluso sacarla del sistema. El arquitecto del sistema nos solicita que la fotocopidora no utilice listas para poder imprimir las guías, ya que buscar una guía en una lista tiene un costo computacional muy alto.

También el nuevo sistema tiene que ser lo suficientemente estable y seguro. Es decir, cada curso siempre debe tener una guía válida, lo mismo sucede con la fotocopidora.

Aclaraciones:

1. Se debe hacer los cambios necesarios para poder implementar el nuevo sistema de la fotocopidora.
 - a. Crear las clases necesarias para poder implementar el sistema, considerando las buenas prácticas vistas en clase
2. Se deberá agregar al main los casos necesarios que prueben que nuestro sistema funciona correctamente.
3. Se evaluará muy detalladamente la solución del sistema así como el entendimiento del enunciado.

ACLARACIÓN:

Como se está trabajando con listas, eliminando y agregando objetos, deberás implementar el equals a las clases que sean necesarias. Además, si querés podés implementar el toString() de estos objetos.