

Personal

Implementa las siguientes clases:

- Persona
- Personal
- Auxiliar
- TestPersonal

Clase Persona:

Clase formada por los siguientes atributos, nombre, apellido, email, género, nacimiento y país. Correspondes con los datos del fichero personal.csv

Se pide implementar los siguientes métodos: constructor, getters y setters, toString, equals

Mas los métodos necesarios para conseguir la funcionalidad pedida

Clase Personal

Clase formada por una colección de objetos del tipo Persona

Se implementarán los métodos siguientes:

- constructor
- addPersona: recibe por parámetro una Persona y la añade a la lista
- borrarPorEmail: recibe un email y borra la persona con dicho email
- personasPais: recibe el nombre de un país y devuelve una lista con todas las personas de ese país
- Mas los métodos necesarios para conseguir la funcionalidad pedida

Clase Auxiliar

En ella implementaremos los siguientes métodos estáticos:

- leerPersonas: recibirá como argumento la ruta (path) del archivo csv a importar y devolver una lista de Personas. El archivo para importar contiene en cada línea los datos de una persona separados por una coma “,”
- guardarPersonas que tendrá como argumento una lista de objetos Persona y creará un fichero cuyo contenido será una línea con cada uno de los datos de un objeto tipo Persona de la lista, separado por comas. Este fichero tendrá el nombre personal_fecha_hora donde la fecha y la hora es la fecha y hora de creación de este fichero.

Clase TestPersonal

Contendrá el main de la aplicación.

Al arrancar debe leer el fichero personal.csv y crear una instancia de la clase Personal con los datos leídos.

Mostrar un menú con las siguientes opciones:

0. Salir

1. Pedir el nombre de un país y mostrar los datos de todas las personas de ese país.

2. Borrar una persona, pidiendo por teclado el mail de esta.

3. Pedir día y mes y mostrar todas las personas que cumplen años ese día

4. Mostrar la persona (o personas) más joven(es) de todas. Utilizar una lista (no un ArrayList)

El programa debe controlar todos los errores posibles usando excepciones (de Java o propias)