
EJERCICIOS PROGRAMACION III

Práctica SEMANA 12

CONTENIDOS: Excepciones y colecciones II: Ordenación y búsqueda - Comparator

Ejercicios obligatorios:

1. Desarrolla una **excepción personalizada** llamada `MyCustomException` siguiendo lo indicado en el guión de prácticas. Haz una que sea checked y unchecked.
2. Realiza un factory method que, en lugar de devolver null en el caso de que no pueda crear un objeto con la información proporcionada, lance (usando **throw**) una excepción que debe ser **checked**.
3. Tomando como base los ejercicios de anteriores sesiones (e.g. ejercicio semana 9), en el apartado de exportación/importación, debes conseguir que, en lugar de devolver un valor predeterminado como retorno en el caso de que se dé una excepción, se propague la excepción (o excepciones) al controlador y a su vez este la propague a la vista, siendo la vista la que controle la excepción y que notifique al usuario de formas distintas dependiendo de la excepción (bloque catch).
4. Tomando como base los ejercicios anteriores (e.g. ejercicio semana 9), debes crear una excepción personalizada y hacer que los métodos de `IRepository` lancen dicha excepción. **Esta excepción deberá envolver las excepciones que se produzcan** e indicarla como causa, por ejemplo, al guardar información en un fichero JSON (o en cualquier otra clase que implemente `IRepository`).
5. Modifica los ejercicios de sesiones anteriores (e.g. ejercicio semana 9) para que utilicen **try catch with resources** en lugar del try catch habitual.
6. Partiendo de uno de los ejercicios de sesiones anteriores (e.g. ejercicio semana 9) haz una opción extra en la que se muestre un listado ordenado por **2 criterios (primero autor y luego año por ejemplo)**. Para ello se deberá crear un **Comparator<T>** y utilizar `Collections.sort` con la sintaxis indicada.
7. Partiendo de uno de los ejercicios de sesiones anteriores, añade una opción que permita buscar en la colección empleando una **búsqueda binaria**. Recuerda que para ello **la colección debe estar previamente ordenada**.

Ejercicios de extensión:

8. **Para mejorar la sintaxis de tu programa, utiliza la Stream API y lambdas en Java para reducir la verbosidad del código java.** Ejemplo: añade una opción a la aplicación que solicite al usuario una palabra y que devolverá un listado de los libros que contienen dicha palabra en su título. **Debes usar lambdas y Stream API para este propósito.**