PRÁCTICA 4 de JAVA:

Tipos genéricos, serialización, excepciones, interfaz y ordenación

Fecha tope entrega	Miércoles, 20 de octubre.	**Fecha tope defensa**	Miércoles, 27 de octubre.
Tipo	Parejas	**Asunto e-mail**	JAVA04
Formato fichero	Apellido1_Nombre1_Apellido2_Nombre2_JAVA04.zip (por classroom)		

Requisitos:

1.- Modelo. Partiendo de la clase **Empleado** desarrolla dos subclases que hereden de ella. Características:

1.1.- Empleado tendrá los siguientes atributos: número (entero), nombre (alfanumérico, además sera **transient** o no serializable por seguridad), sueldo y sueldo máximo (real para cada empledo) y fecha de alta (fecha).

1.2.- La subclase **Analista** tendrán los atributos:

- Plus económico anual (real, se expresará en tanto por ciento)
- Otro que debes personalizar.

1.3.- La subclase **Programador** tendrán los siguientes atributos:

- Sueldo extra mensual (real).
- Otro a personalizar, pero distinto en nombre y tipo al de Analista

2.- Modelo. Desarrolla un **interfaz e implementarse sobre las clases hijas** en los cálculo con fechas. Deberá:

2.1.- Tener los métodos para controlar si se cumplen meses, trimestres o años. Dichos métodos no tendrán parámetros y compararán la fecha de apertura de la Empleado y la fecha del sistema.

2.2.- Tendrá las constantes de la clase Calendar (DAY_OF_MONTH, MONTH y YEAR) en **español**.

3.- Molelo. Desarrollo un **sistema de excepciones** para controlar que el sueldo de cada empleado nunca supere el sueldo máximo. Su control debe ser jerarquizado (try, catch, …, catch y finally). Los mensajes de las excepciones solo serán visibles en la vista del MVC.

4.- Controlador. Debes utilizar las clases **Lista** y **Nodo** de la práctica anterior, con modificaciones para que tenga un **índice** (numero de empleado), que almacene **tipos genéricos** y desarrollando los **métodos que creas necesarios**, como por ejemplo intercambiar(x,y)).

5.- Controlador. Los objetos se cargarán desde un **fichero**, si existe, a una lista (puede estar vacía). Y el proceso inverso: se podrán salvar o guardar desde la lista a fichero. Debe usarse el intefaz **Serializable**.

6.- Vista. Implementa una aplicación gráfica con las siguientes opciones:

6.1.- **Cargar** datos del fichero a lista.

6.2.- **Guardar** datos de lista al fichero.

6.3.- **Insertar** o nuevo empleado (**analista o programador)**. El número de empleado será inferior a 100 (los aleatorios serán superiores). Para la defensa habrá objetos con datos para que cumplan las condiciones de tiempo para realizar cálculos sobre el sueldo (mes o trimestre) y salte la excepción sobre el sueldo máximo.

6.4.- Visualizar en **JList** todos los objetos indicando el tipo de cada uno (Analistas o Programador).

6.5.- **Visualizar uno a uno** los elementos, con la posibilidad de operar solo el empleado mostrado. Tendrá los botones anterior, siguiente y **calcular** (el sueldo con el plus o el sueldo extra).

6.6.- El botón **calcular solo estará activo sí se cumple los periodos** (mes para Programadores o años para Analistas). Modificará el sueldo si procede e informando de lo ocurrido con detalle.

7.- Una opción gráfica también permitirá **ordenar** la lista con los siguientes pasos:

7.1.- Primero completará la lista con 25.000 (¿?) empleados, generado el atributo número de forma aleatoria y siendo mayor a 1000.

7.2.- Seguidamente, implementa una **collections** y copiará la lista original en ella.

7.3.- Finalmente mide el tiempo en ordenar cada una de las estructuras de datos por el atributo número. Se mostrarán los tiempos en mili-segundos y los 100 primeros empleados por consola.

7.4.- Tras ordenar se podrán hacer un listado por consola, o mejor en el Jlist.

8.- Controlador. Clase para comprobar mediante excepciones. que cada uno los atributos de un nuevo empleado (por formulario) son correctos (Date validated. etc).

9.- Estará estructura siguiendo el MVC, también los paquetes.

10.- Mejoras: Date Picker, Filechooser, refresco actualizaciones, control de errores, Jlist seleccionable, LookAndFeel, manejo de fechas, etc.

Ayuda:

- Serializable:
  - http://chuwiki.chuidiang.org/index.php?title=Serializaci%C3%B3n_de_objetos_en_java
  - https://www.geeksforgeeks.org/serialization-in-java/
  - https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html
- Generics:
  - https://docs.oracle.com/javase/tutorial/extra/generics/index.html
  - https://www.journaldev.com/1663/java-generics-example-method-class-interface
- Collections.sort:
  - https://www.geeksforgeeks.org/collections-sort-java-examples/
- LookAndFeel:
  - http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

Ejemplos prácticos:

- http://es.wikihow.com/serializar-un-objeto-en-Java
- http://www.chuidiang.com/java/ficheros/ObjetosFichero.php