

Práctica 2: Genericidad, excepciones y Entrada/Salida

1. Objetivo

En esta práctica vas a utilizar Genericidad, además de algunas funcionalidades de Java para la definición de excepciones propias y la serialización.

2. Enunciado

En esta práctica vas a añadir funcionalidad a tu agenda para que sea capaz de devolverte los clientes/facturas/llamadas registrados en el sistema entre dos fechas, para ello utilizarás Genericidad

Además, definirás excepciones que describan las situaciones anómalas en la ejecución de tu aplicación, por ejemplo, que la búsqueda de un usuario a partir de su nif no da como resultado ningún usuario registrado en el sistema.

Finalmente harás uso de la serialización para almacenar y recuperar los datos de tu agenda en formato binario.

3. Metodología

Esta práctica la desarrollarás en dos sesiones.

Sesión 1 (Genericidad)

Genericidad

En el enunciado de la práctica se pidió que las clases Cliente, Llamada y Factura dispusieran de un método llamado `getFecha()`. El objetivo era poder tratar todas estas clases de manera similar a la hora de buscar objetos que correspondan a un periodo entre dos fechas. Para ello emplearemos el mecanismo de la Genericidad.

En esta sesión, te pedimos que escribas un método genérico que reciba tres argumentos:

- Un conjunto de objetos. Cuando decimos conjunto, queremos decir alguna estructura de datos de las que dispone Java y que contenga objetos. En nuestro caso, como el código de la función que tendrás que escribir necesitará consultar de cada objeto la función `getFecha()`, estaremos hablando de que la función, en la práctica, recibirá un conjunto de clientes, llamadas o facturas.
- Una fecha de inicio.
- Una fecha final.

El objetivo de la función será devolver un conjunto de objetos, extraído del conjunto que se pasó como argumento, donde los objetos seleccionados cumplan la restricción de que su fecha se encuentra entre las fechas inicial y final pasadas como argumentos.

Añade a tu programa tres opciones:

- Mostrar un listado de clientes que fueron dados de alta entre dos fechas.
- Mostrar un listado de llamadas de un cliente que fueron realizadas entre dos fechas.
- Mostrar un listado de facturas de un cliente emitidas entre dos fechas.

Sesión 2 (Excepciones y Entrada/Salida)

Excepciones

Durante la ejecución de toda aplicación se pueden producir situaciones excepcionales. Estas situaciones excepcionales las debes codificar como clases. Analiza cuáles son las situaciones excepcionales que se pueden dar durante la ejecución de tu aplicación y escribe las excepciones necesarias para gestionarlas. Como ejemplo, al definir el período correspondiente a una factura, la fecha de inicio no puede ser posterior a la fecha de fin.

Entrada y salida de datos

En esta sesión debes añadir las dos siguientes opciones a tu programa:

- Guardar todos los datos de la aplicación en un fichero al cerrar la aplicación.
- Cargar todos los datos desde un fichero al iniciar la aplicación.

4. Fuentes de información

- [Big Java](#) Capítulos 2, 8 y 9.
- [Desarrollo de proyectos informáticos con tecnología Java](#) Capítulo 3.
- [JUnit](#) Página web de JUnit. Allí encontrarás los javadocs de este framework.