



- Tarea UT08 - Colecciones de datos

Enunciado

Ejercicio 1:

Diseña un programa en Java para gestionar una biblioteca digital de libros. El programa debe permitir a los usuarios realizar operaciones como agregar libros a la biblioteca, buscar libros por título y mostrar todos los libros disponibles. Para esto, utilizarás la estructura de datos HashMap para almacenar la información de los libros.

- El programa debe permitir al usuario ingresar los detalles de un libro, incluyendo título, autor, año de publicación, género y número de páginas.
- Después de ingresar los detalles del libro, el programa debe agregarlo a la biblioteca digital utilizando un HashMap. Utiliza el título del libro como clave y un objeto Libro como valor, dónde Libro es una clase que contendrá los detalles mencionados anteriormente.
- Implementa una función que permita al usuario buscar un libro por su título. Si el libro está en la biblioteca, se debe mostrar toda la información relacionada con ese libro.
- Implementa una función que muestre todos los libros disponibles en la biblioteca digital, mostrando tanto el título como los detalles completos de cada libro.

Clase Libro:

Atributos:

- titulo (String): El título del libro.
- autor (String): El autor del libro.
- fechaPublicacion (LocalDate): El año de publicación del libro.
- generoLiterario (String): El género del libro.
- numPaginas (int): El número de páginas del libro.
- editorial (String): La editorial que publica el libro.

Crea un hashMap<string, libro> para almacenar los libros en la biblioteca digital. Crea un método main para probar y ejecutar todo lo pedido en el ejercicio.

Ejercicio 2:

Crea un programa que pida por teclado un número entero positivo N y cree un conjunto ordenado de los números primos inferiores o iguales a N. Utilizando el tipo de conjunto más apropiado para ello. Crea un método main para probar y ejecutar todo lo pedido en el ejercicio. Deberá mostrar por pantalla el conjunto.

Ejemplo: Si N = 31

C={2, 5, 7, 11, 13, 17, 19, 23, 29, 31}

Ejercicio 3:

Desarrolla un programa que sea un Gestor de Tareas Diarias en Java, que te ayude a planificar y hacer seguimiento de lo que tienes pendiente y lo que ya has completado.

El programa deberá permitir las siguientes opciones que se mostrarán al usuario para seleccionarlas mediante un menú:

- Agregar nuevas tareas: Las tareas pendientes se almacenarán en una lista de tareas por hacer (ArrayList<String>).
- Marcar tareas como completadas: Una vez completada una tarea, deberá moverse a la lista de tareas realizadas (LinkedList<String>).
- Consultar el estado actual de las tareas: Se debe mostrar claramente cuáles están pendientes y cuáles ya han sido completadas.
- Eliminar varias tareas a la vez: Si varias tareas dejan de ser necesarias, se debe poder eliminarlas en bloque usando subList().

- Recuperar la última tarea completada: Si una tarea fue marcada como completada por error, se debe poder restaurar a la lista de pendientes utilizando poll().
- Modificar una tarea existente: En caso de que nos hayamos equivocado al nombrar la tarea o queramos especificar algo más. Utilizando indexOf().

Criterios de puntuación. Total 10 puntos

- Ejercicio 1: 4 puntos
- Ejercicio 2: 2.5 puntos
- Ejercicio 3: 3.5 puntos
- Todos los archivos .java deben estar correctamente comentados y estructurados. Se penalizará con hasta 2 puntos por este motivo.
- Si la tarea no cumple con el formato, no compila o alguna de las instrucciones de entrega no se considerará válido y se evaluará con un 0.

Indicaciones de entrega

Cada ejercicio estará contenido en un único fichero cuyo nombre sea similar a **PROG08_EjerXX**, donde la XX se sustituirá por el número del ejercicio. El tipo de archivo a entregar dependerá del ejercicio:

Los archivos fuentes de cada ejercicio (no el proyecto completo, sólo las fuentes).

Asegúrate de que los archivos fuente (.java), contengan, como comentarios Java, tu nombre y apellidos y el número de ejercicio.

Los archivos fuente deben compilar y ejecutarse siguiendo las instrucciones de su enunciado. De no hacerlo, no se considerarán válidos y serán evaluados con un 0.

En el supuesto que tengas/quieras adjuntar un documento de texto con explicaciones adicionales, o con la solución a alguno de los puntos pedidos, debes de seguir las siguientes consideraciones que se explican en el siguiente [documento](#).

Crea una carpeta para la entrega y adjunta los archivos creados, y renombra la carpeta siguiendo las siguientes pautas:

apellido1_apellido2_nombre_PROG_Tarea08

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños.

IMPORTANTE

Si la carpeta es muy pesada para el límite máximo de subida de la plataforma Moodle, súbela a tu unidad del Drive. Utiliza la cuenta de Educación, y en la entrega en Moodle adjunta la URL de tu entrega.

Para que la entrega se considere realizada en tiempo y forma, sube al Moodle el PDF con la solución de tu tarea, y adjunta el resto de los archivos que has creado al Drive, y comparte la carpeta mediante su enlace correspondiente. **No olvides ajustar los privilegios de la carpeta en Drive para que pueda acceder a tu trabajo.**

Todos los ejercicios y/o /documentos demandados que no cumplan exactamente con las instrucciones y su formato de entrega no se considerarán válidos y serán evaluados con un 0.