UTNMDP
Regional Mar del Plata

Guia #7
JSON

1. Crea una clase Persona con los siguientes atributos: nombre, edad, dni, y sexo. Implementa un constructor, getters y setters para los atributos.
   Ahora, implementa un método que convierta un objeto Persona a una representación JSON y luego guarda la representación JSON en un archivo llamado persona.json. Elige y utiliza una biblioteca de tu elección para la serialización de objetos a JSON y la escritura en archivos.

   Objetivos:
   - Convertir un objeto Java a JSON.
   - Escribir datos JSON en un archivo.

2. Utiliza la clase Persona del ejercicio anterior. Lee el archivo persona.json que creaste y convierte el JSON leído de nuevo a un objeto Persona.

   Objetivos:
   - Leer datos JSON desde un archivo.
   - Convertir JSON a un objeto Java.

3. Crea una clase Curso con atributos como nombreCurso, código, y una lista de Personas inscritas en el curso. Implementa métodos para agregar y eliminar personas del curso.
   Luego crea un método que convierta un objeto Curso a JSON y guárdalo en un archivo curso.json.
   Implementa otro método que lea el archivo curso.json y convierta el JSON a un objeto Curso.

   Objetivos:
   - Trabajar con listas de objetos y su persistencia en JSON.
   - Leer y escribir JSON que contenga una estructura compleja.

4. Crea una clase Biblioteca con atributos como nombreBiblioteca y una lista de libros e implementa métodos para agregar y eliminar libros.
   Guarda el estado actual de la biblioteca en un archivo biblioteca.json.
   Lee el archivo biblioteca.json, realiza cambios en la lista de libros (añade o elimina algunos) y guarda los cambios de nuevo en el archivo.

Objetivos:

- Manipular datos en JSON para actualizar información existente.
- Leer y escribir JSON con modificaciones.

5. Crea una clase Empleado con los atributos id, nombre, salario, y departamento.
   Implementa métodos para convertir un objeto Empleado a JSON y viceversa.
   Agrega validaciones para asegurarte de que el JSON que se lee tiene el formato correcto y contiene todos los campos necesarios.
   Maneja los posibles errores que puedan ocurrir durante la lectura y escritura de archivos JSON, como archivos no encontrados o datos mal formateados.

   Objetivos:

   - Implementar validación de datos JSON.
   - Manejar errores y excepciones durante el procesamiento de archivos JSON.