

TAREA PARA PROG06

Detalles de la tarea de esta unidad.

Introducción.

En esta sexta unidad de trabajo del módulo has aprendido las principales operaciones que se pueden realizar con ficheros secuenciales y de acceso aleatorio en Java, y la serialización de datos.

Hay diferentes realizar la aplicación cumpliendo los objetivos de uso de ficheros con serialización de datos (ojo, esto es imprescindible).

Se da cierta flexibilidad y podéis basaros en el ejercicio de ejemplo que está en la plataforma, dentro de la unidad.

Enunciado.

Se trata de hacer una aplicación en Java para almacenar, mantener y consultar datos sobre **cuerpos celestes del sistema solar**. Esos datos, se almacenarán en un fichero **serializado**, denominado **sistemasolar.dat**.

Habrás 2 paquetes:

- uno llamado **modelo** en el que irá la clase **CuerpoCeleste.java**. Esta clase no tendrá main (no lo necesita), será **serializada** y contendrá los **atributos, métodos y constructores** que consideres necesarios, basándonos en las unidades anteriores. Además tendrá el **toString()** que transforme en una cadena el valor de los atributos del objeto en ese instante.
- Otro paquete, llamado **programacuerposcelestes** en el que irán el resto de clases y métodos necesarios para la operativa del programa. Además, aquí irá el main.
En este paquete puede haber una sola clase con el main y resto de métodos para hacer toda la operativa, o bien, varias clases, según te estructures (una para utilidades que lea de pantalla, por ejemplo, otra para el programa en sí).
El caso es que se inicializará el programa, mostrando la bienvenida y el menú y realizará toda la operativa del programa utilizando la clase **CuerpoCeleste** que habrá que haber importado.)

Los datos que se almacenarán sobre cada cuerpo son (**CuerpoCeleste.java**):

- **codigoCuerpo**: Numérico entero de 3 dígitos. Se asigna libremente por parte del usuario.
- **nombre**: Cadena de texto, tamaño máximo 15.
- **tipoObjeto**: Cadena de texto. No es obligatorio controlar que el valor introducido es uno de los tipos. Los tipos pueden ser, "planeta", "planetaEnano", "luna"
- **diametro**: Número entero, 6 dígitos. Significa su valor de diámetro en kilómetros, no es necesario decimales.

Elige el **tipo de datos más adecuado** para cada tipo. No es necesario, de cara a la evaluación, validar el tamaño de cada dato, así como que se introduce un tipo de objeto de los dados arriba, pero si lo realizas es un programa más completo.

Recuerda añadir **constructores**, métodos **getters** y **setters** necesarios, y el **toString()** como se indicaba anteriormente.

***nota:** en astronomía, se llama luna de manera genérica a cualquier satélite natural que tienen los planetas

Ejemplo de algunos datos:

codigoCuerpo	nombre	tipoCuerpo	Diametro
001	Júpiter	planeta	142984
002	Marte	planeta	6787
101	Ceres	planetaEnano	952
102	Plutón	planetaEnano	2370
201	Luna	luna	3470
202	Ganímedes	luna	5262
203	Encélado	luna	499

*en el ejemplo se sigue una codificación, pero para tu ejercicio puedes seguir esa misma u otra.

En los siguientes enlaces tenéis los principales cuerpos del sistema solar por si queréis hacer pruebas realistas.

Planetas: [https://es.wikipedia.org/wiki/Sistema_solar_-_~:text=no lo está,-Características principales,-Artículo principal%3A Anexo%3ADatos](https://es.wikipedia.org/wiki/Sistema_solar_-_~:text=no%20lo%20está,-Características principales,-Artículo principal%3A Anexo%3ADatos)

Planetas Enanos: https://es.wikipedia.org/wiki/Sistema_solar_-_~:text=eclíptica *** [Satélites naturales,-Planetas enanos,-Los cinco planetas](#)

Lunas: https://es.wikipedia.org/wiki/Sistema_solar_-_~:text=de la Tierra,-Grandes satélites del sistema solar,-Véase también%3A Anexo%3ALista

Funcionamiento del programa:

Al entrar en la aplicación se mostrará un mensaje de inicio, con datos del nombre del programa, para que el usuario sea consciente de que ha empezado, y a continuación el siguiente menú.

Mediante el menú se podrán realizar determinadas operaciones:

1. **Añadir cuerpo celeste.** Esta opción pedirá los datos del cuerpo celeste y añadirá el registro correspondiente en el fichero. Si existe se añade al fichero, y si no, el fichero se crea y se añade. Se mostrará un mensaje por consola (System.out) diciendo que se ha añadido correctamente o que no se ha podido añadir según sea el caso.
2. **Listar cuerpos celestes.** Recorrerá el fichero mostrando los datos de los cuerpos almacenados en el mismo. (que haya una separación entre cuerpo y cuerpo). Si no hay datos se mostrará en un mensaje.
3. **Buscar cuerpo celeste por código.** Pedirá al usuario el código del cuerpo a buscar, y comprobará si existe en el fichero, mostrando los datos de dicho cuerpo en caso afirmativo y dando un mensaje de que no existe si encuentra ese código.
4. **Buscar cuerpo celeste por tipo.** Pedirá al usuario el tipo de cuerpo celeste que se quiere buscar. Si hay cuerpos de ese tipo mostrará todos los cuerpos de ese tipo, por ejemplo, todas las lunas, (separando entre cuerpo y cuerpo), si no se indicará en un mensaje que no existen cuerpos celestes de ese tipo.
5. **Borrar cuerpo celeste.** Pedirá al usuario el código del cuerpo a borrar, y si existe, lo borrará del fichero. Si no, se indicará un mensaje de que no existe.
6. **Borrar fichero de cuerpos celestes.** Elimina del disco el fichero sistemasolar.dat indicando en un mensaje si se ha podido borrar o si ha habido algún error.
7. **Salir de la aplicación.** Se mostrará un mensaje de despedida.

La aplicación tiene que tener un control de excepciones apropiado (al menos controlar las excepciones de *FileNotFoundException*, de *IOException* y una genérica o *Exception*)

Criterios de puntuación. Total 10 puntos.

Requisito para evaluar, que el documento demuestre la identificación del alumno, mostrando de fondo en los pantallazos, el aula virtual del curso junto con la identidad.

- Inicio, mostrar menú y funcionamiento del menú con petición y entrada correcta de datos, opción de salir. **(1 punto)**
- Cada opción del menú del 1 al 6. (1 punto, total **6 puntos**)
- Documento PDF, bien elaborado, con pantallazos demostrando el funcionamiento de todas las opciones del menú una a una, explicando cómo se ha acometido el programa, sin errores de compilación, y que la ejecución del programa sea legible y clara. **(1 punto)**
- Control de excepciones correcto. **(1 punto)**
- Estructura del programa, con comentarios, indentación, clases separadas en paquetes, elección de tipos, ausencia de errores, etc. **(1 punto)**

Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- Sistema operativo Windows o Linux.
- JDK y JRE de Java (preferiblemente últimas versiones)
- NetBeans IDE 6.9.1 o superior. (preferiblemente Apache NetBeans 12.5)

Consejos y recomendaciones.

Dentro de la carpeta de NetBeans, se habrá creado automáticamente la carpeta del proyecto. Comprueba que efectivamente está la carpeta y dentro todo el contenido que has creado.

Para entregar, **comprime en un único** archivo la **carpeta del proyecto** con todo su contenido, junto con el **documento PDF y pantallazos con las explicaciones que demuestran la identidad**.

Indicaciones de entrega.

Se enviará un **único archivo comprimido que contendrá el proyecto, y un documento PDF**, con pantallazos que muestren identidad y explicaciones necesarias indicadas más arriba, sobre el programa.

En ese documento se tiene que explicar cómo se ha realizado el programa, y dar pantallazos de todas las opciones funcionando y explicando lo necesario.

El envío se realizará a través de la plataforma y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas para la quinta unidad del MP de PROG, debería nombrar esta tarea como:

sanchez_manas_begona_PROG06_Tarea.