

Tarea 4 – Set

Enunciado

Crea el siguiente programa:

1. `CuerpoCeleste` (clase)
 - a. Un enumerado interno anidado denominado `TipoCuerpoCeleste` declarada como pública con los siguientes valores: `ESTRELLA`, `PLANETA`, `PLANETA_ENANO`, `LUNA`, `COMETA`, `ASTEROIDE`.
 - b. Tiene los siguientes campos privados: `nombre`(String), `periodoOrbital`(double), `satélites`(Set<CueporCeleste>), `tipoCuerpo` (TipoCuerpoCeleste).
 - c. Un constructor que toma un String (nombre), un doble (su período orbital) y un tipoCuerpo. Los satélites se inicializan con un nuevo HashSet.
 - d. Método `getPeriodoOrbital`, `getNombre`, `getTipoCuerpo`.
 - e. `getSatelites`, devuelve un nuevo HashSet con la copia del set de satélites.
 - f. `addSatelite`, recibe un `CuerpoCeleste` como parámetro y devuelve verdadero si se agregó con éxito, falso de lo contrario.
 - g. `equals`. Dos objetos `CuerpoCeleste` son iguales si su nombre y tipo son iguales y son de tipo clase `CuerpoCeleste`.
 - h. `hashCode`. Devuelve el código hash del nombre + el código hash del tipo + un número arbitrario.
 - i. `toString`. Devuelve la representación de String como: "nombre: tipo, periodoOrbital".
2. `Planeta` (clase hija)
 - a. Un constructor que toma un String (nombre del planeta) y un doble (su período orbital) y llama a su clase principal con sus argumentos y se especifica el tipo de cuerpo.
 - b. `addSatelite`. Agrega el satélite si su tipo de cuerpo es una luna.
3. `PlanetaEnano` (clase hija)
 - a. Un constructor que recibe un String (nombre del planeta enano) y un doble (su período orbital) y llama a su clase principal con sus argumentos y con el tipo de cuerpo.
4. `Luna` (clase hija)
 - a. Un constructor que toma un String (nombre de la luna) y un doble (su período orbital) y llama a su clase principal con sus argumentos y con el tipo de cuerpo.
5. `Main` (clase)
 - a. Crea un atributo estático de tipo Map<String, CuerpoCeleste> llamado `sistemaSolar` que sea un HashMap.
 - b. Crea un atributo estático de tipo Set<CuerpoCeleste> llamado `planetas` que sea un HashSet.
 - c. Crea un método `main` con:
 - i. Crea un objeto `CuerpoCeleste` de tipo `Planeta` con Mercurio y periodo orbital 88. Añade ese objeto al `sistemaSolar` y mételo también dentro del conjunto de `planetas`.
 - ii. Haz lo mismo para el planeta Venus-225, La Tierra-365, Marte-687, Jupiter-4332, Saturno-10759, Urano-30660, Neptuno-165, Pluton-248.

PROGRAMACIÓN

UD6. Colecciones

- iii. Crea un objeto `CuerpoCeleste` de tipo Luna y periodo orbital 27, agrégalo al `sistemaSolar` y añade al objeto `LaTierra` que creamos en el apartado anterior este satélite.
- iv. Crea otras dos lunas más (Deimos-1,3 y Phobos-0,3) añádelas al sistema solar y agrégalas también en el planeta Marte como satélites.
- v. Crea otras cuatro lunas más (Io-1,8 Europa-3,5, Ganymede-7,1 Callisto-16.7) añádelas al sistema solar y agrégalas también en el planeta Júpiter como satélites.
- vi. Con un `for-each` imprime el nombre de todos los planetas que hemos almacenado en el conjunto (`Set`).
- vii. Obtén del mapa del sistema solar el planeta Marte e imprime todas sus lunas.
- viii. Vamos a obtener la unión de todas las lunas, para eso nos creamos un `Set<CuerpoCeleste>` lunas de tipo `HashSet`. Y con un `for-each` recorremos todos los planetas y hacemos la unión de los satélites para cada uno de los planetas.
- ix. Imprime el set de lunas del apartado anterior, debería haber 7 lunas.
- x. Crea un planeta Plutón-884. Imprime de nuevo los planetas utilizando el método `sobreescrito toString`. ¿Se ha agregado este planeta al conjunto? ¿Por qué?
- xi. ¿Y si lo creamos como un planeta enano en vez de un planeta?
- xii. Utiliza la diferencia y la intersección de `Set` utilizando el código. Añade lo que necesites de código y explica también qué tenemos en los conjuntos y cuál es el resultado esperado al aplicar las operaciones.

Entrega

- Sube un PDF con el código copiado.