SEGUNDO PROYECTO JAVA

En este proyecto se repasarán los conceptos vistos durante el primer proyecto, con el fin de reforzar los contenidos vistos y tratados. Además se trabajarán nuevos conceptos entre los que destacarán:

- Interfaces en Java
- Lectura, escritura y tratamiento de ficheros en Java.
- Lectura y transformación de ficheros XML a objetos Java.
- Conexión a BBDD en Java.
- Extracción y transformación de información de BBDD en Java.
- Repaso general de conceptos (Herencia, Polimorfismo, Abstracción, Accesibilidad, etc).

Requisitos V5

1. Se ha de crear un proyecto Maven, llamado ***proyectoLeagueOfLegends***.
2. En el paquete src >main> java se ha de crear un nuevo paquete llamado ***modelos***.
3. En la carpeta modelos se han de crear los siguientes clases:

Personaje

- id -> int
- nombre -> String
- descripcion -> String
- fechaCreacion-> LocalDate
- nivel -> Integer
- vidaBase-> Double
- manaBase -> Double
- defensaBase -> Double
- defensa ->Double
- ataqueBase -> Double
- ataque -> Double
- vida-> Double
- mana -> Double
- Region -> Enum (DEMACIA, NOXUS, JONIA, SHURIMA, TARGON, FREIJORD, PILTOVER, ZAUM, BANDLE, AGUAS_ESTANCADAS, DESCONOCIDA)
- List<Habilidad> habilidades
- List<Item> equipamiento
- Escalabilidad escalabilidad

Habilidad

- id-> int
- nombre-> String
- dañoBase -> Double
- daño -> Double
- costeMana -> Double
- TipoHabilidad-> Enum (DAÑO, CURACION, BUFO_VIDA, BUFO_DEFENSA, BUFO_PODER)

Item

- id-> int
- nombre-> String
- aumentoDaño -> Double
- aumentoDefensa-> Double
- aumentoSalud-> Double
- aumentoMana-> Double

Escalabilidad

- id-> int
- incrementoDañoNivel -> Double
- incrementoDefensaNivel-> Double
- incrementoSaludNivel-> Double
- incrementoManaNivel-> Double

Todos los modelos han de contar con los atributos privados que se detallan , al igual que el constructor vacío, copia y completo, los métodos get y set de todos los atributos y los métodos toString, equals y hashcode.

4. En el paquete src >main> java > se ha de crear un nuevo paquete llamado ***utilidades***.
5. En el paquete src>java>utilidades se ha de crear las siguientes clases:
    a. UtilidadesPersonaje.class
    b. UtilidadesHabilidad.class
    c. UtilidadesItem.class

6. En la clase UtilidadesPersonaje se han de implementar los siguientes métodos .

.

- public Personaje levelUp(Personaje personaje)
    - Que recibiendo un personaje aumente su nivel en 1, actualizando todas sus estadísticas según su escalabilidad.
        - Estadística = EstadísticaBase + EcalabilidadEstadística * nivel (Tras la subida).

- public Map<Region, List<Personajes> getPersonajesPorRegion(List<Personaje> personajes)
    - Que a partir de un listado de personajes devuelve un mapa de los personajes agrupados por región.

- public Personaje getMasPoderoso(List<Personaje> personaje)
    - A partir de una lista de personajes, devuelve el personaje más poderoso. Para saber si un personaje es más poderoso que otro, hay que:
        - Levear a nivel 18 (Independientemente del nivel en el que esté).
        - Actualizar sus stats según el nivel 18.
        - El personaje que tenga su suma de estadísticas más altas a nivel 18 será el más poderoso.

- public Map<Region, List<Personajes> getMasPoderosoPorRegion(List<Personaje> personajes)
    - Que devuelve el personaje más poderoso de cada región.