

Especificació capa persistència

- Classe PBoard: S'encarrega de gestionar la informació dels taulers al disc.

- Mètodes:

`public static String read(String name, String user) throws IOException`

Mètode estàtic que s'encarrega de llegir un tauler del disc. Rep com a paràmetres el nom del tauler i el nom de l'usuari que l'ha creat. Retorna una string amb el contingut de les 64 caselles del tauler ordenades per files. Llança una `IOException` si hi ha cap problema llegint el tauler o si el format d'aquest és incorrecte i elimina el tauler conflictiu.

`public static void write(String name, String content, boolean overwrite, String user) throws ConfirmationException, IOException`

Mètode estàtic per a guardar el contingut d'un tauler al disc. Rep com a paràmetres el nom del tauler, els continguts d'aquest, un booleà indicant si volem sobre escriure el tauler en cas de que existeixi i el nom de l'usuari que ha creat el tauler. Els continguts són guardats al tauler del nom indicat pertanyent a l'usuari indicat a menys que el tauler ja existeixi i `overwrite` sigui fals. Llança una `IOException` si hi ha cap problema en l'escriptura i una `ConfirmationException` si el tauler ja existeix i `overwrite` és fals.

`public static void delete(String name, String user)`

Mètode estàtic per a esborrar un tauler existent. Rep com a paràmetres el nom del tauler i el nom de l'usuari que l'ha creat. Rere la execució, el tauler indicat per nom pertanyent a l'usuari indicat per `user` no existeix.

- Classe PGame: S'encarrega de gestionar la informació de les partides al disc.

- Mètodes:

`public static String read(String name, String user) throws IOException`

Mètode estàtic que s'encarrega de llegir una partida del disc. Rep com a paràmetres el nom de la partida i el nom de l'usuari que l'ha guardat. Retorna una string amb tots els atributs de la partida. Llança una `IOException` si hi ha cap problema llegint la partida o si el format d'aquesta és incorrecte i elimina la partida conflictiva.

`public static void write(String name,String content, boolean overwrite, String user) throws ConfirmationException,IOException`

Mètode estàtic per a guardar la informació d'una partida al disc. Rep com a paràmetres el nom de la partida, la informació d'aquesta, un booleà indicant si volem sobre escriure la partida en cas de que existeixi i el nom de l'usuari que ha guardat la partida. La informació és guardada a la partida del nom indicat pertanyent a l'usuari indicat a menys que la partida ja existeixi i `overwrite` sigui fals. Llança una `IOException` si hi ha cap problema en l'escriptura i una `ConfirmationException` si la partida ja existeix i `overwrite` és fals.

`public static void delete(String name, String user)`

Mètode estàtic per a esborrar una partida existent. Rep com a paràmetres el nom de la partida i el nom de l'usuari que l'ha guardat. Rere la execució, la partida indicada per nom pertanyent a l'usuari indicat per `user` no existeix.

- Clase `PRanking`: S'encarrega de gestionar la informació del ranking al disc.

- Mètodes:

`public static String read(String mode) throws IOException`

Mètode estàtic per a llegir un ranking del disc. Rep com a paràmetres la parametrització de ranking desitjada en forma de string. Retorna una string amb la informació del ranking demanat. Llança una `IOException` si hi ha cap problema tractant de llegir el ranking o si el format d'aquest és incorrecte i elimina el ranking conflictiu.

`public static void write(String mode, String content) throws IOException`

Mètode estàtic que s'encarrega d'escriure el contingut d'un ranking en el disc. Rep com a paràmetres la parametrització desitjada de ranking en format string i els continguts d'aquest ranking. El ranking seleccionat passa a tenir els continguts indicats. Llança una `IOException` si hi ha cap problema a l'escriptura.

- Clase `PUser`: S'encarrega de gestionar la informació dels usuaris al disc

- Mètodes:

`public static void create(String name, String password) throws ConfirmationException,IOException`

Mètode estàtic per a enregistrar un nou usuari al disc. Rep com a paràmetres el nom de l'usuari i la seva contrasenya. Si l'usuari no existia, la seva carpeta es creada amb un arxiu on es guarda la contrasenya i dues carpetes per als taulers i les partides guardades. Llança una `ConfirmationException` si l'usuari ja existia i una `IOException` si la creació ha anat malament.

```
public static void delete(String name)
```

Mètode per a esborrar un usuari. Rep com a paràmetre el nom de l'usuari a esborrar. Rere l'execució, l'usuari indicat no existeix.

```
private static void delete_directory(File f)
```

Mètode recursiu per a esborrar els continguts d'un directori abans d'esborrar el directori indicat. Rep com a paràmetre el directori a esborrar. Rere l'execució, el directori indicat no existeix.

```
public static String getPassword(String user) throws IOException
```

Mètode estàtic per a consultar la contrasenya d'un usuari. Rep com a paràmetre el nom de l'usuari. Retorna una string amb la contrasenya d'aquest. Si hi ha cap problema amb la lectura retorna una `IOException` i elimina el fitxer conflictiu.

```
public static String[] getGames(String user)
```

Mètode estàtic per a consultar les partides guardades d'un usuari. Rep com a paràmetre el nom de l'usuari. Retorna un array de strings amb tots els noms de les partides d'aquest usuari.

```
public static String[] getBoards(String user)
```

Mètode estàtic per a consultar els taulers creats d'un usuari. Rep com a paràmetre el nom de l'usuari. Retorna un array de strings amb tots els noms dels taulers d'aquest usuari.

■ Clase Persistence: Controlador de la capa de persistència

- Mètodes

```
public static String readBoard(String name, String user) throws IOException
```

Mètode estàtic que crida al mètode `read` de la classe `PBoard`.

```
public static void saveBoard(String name, String content, boolean overwrite, String user)
throws ConfirmationException,IOException
```

Mètode estàtic que crida al mètode write de la classe PBoard.

```
public static void deleteBoard(String name, String user)
```

Mètode estàtic que crida al mètode delete de la classe PBoard.

```
public static String readGame(String name, String user) throws IOException
```

Mètode estàtic que crida al mètode read de la classe PGame.

```
public static void saveGame(String name, String content, boolean overwrite, String user)
throws ConfirmationException,IOException
```

Mètode estàtic que crida al mètode write de la classe PGame.

```
public static String readRanking(boolean d, boolean h, boolean v) throws IOException
```

Mètode estàtic que crida al mètode read de la classe PRanking. Rep com a paràmetres tres booleans (diagonal, horitzontal i vertical) a partir dels quals extreu la string de la parametrització desitjada que demana la classe PRanking

```
public static void saveRanking(boolean d, boolean h, boolean v, String content) throws
IOException
```

Mètode estàtic que crida al mètode write de la classe PRanking. Rep com a paràmetres tres booleans (diagonal, horitzontal i vertical) a partir dels quals extreu la string de la parametrització desitjada que demana la classe PRanking

```
public static void createUser(String name, String password) throws
ConfirmationException,IOException
```

Mètode estàtic que crida al mètode create de la classe PUser.

```
public static void deleteUser(String name)
```

Mètode estàtic que crida al mètode delete de la classe PUser.

```
public static String getPassword(String name) throws IOException
```

Mètode estàtic que crida al mètode getPassword de la classe PUser.

```
public static String[] getGames(String name)
```

Mètode estàtic que crida al mètode getGames de la classe PUser.

```
public static String[] getBoards(String name)
```

Mètode estàtic que crida al mètode getBoards de la classe PUser.

```
public static void deleteGame(String name, String user) throws IOException
```

Mètode estàtic que crida al mètode delete de la classe PGame.

```
public static boolean userExists(String name)
```

Mètode estàtic que crida al mètode exists de la classe PUser.