Presentación

Hola Abelardo,

Estoy un poco nervioso por que es mi primera entrevista en ingles y mi segunda entrevista para la presentar una aplicación.

Así que para mi, va a ser difícil.

Presentación primera app

Esta es la aplicación de la primera evaluable.

Se trata de una aplicación para almacenar y leer información en CSV y XML de un torneo de ajedrez. Voy a explicarte el código.

Código Main()

Importo todas las librerías necesarias

Creo la clase GestChessPlayers que es la encargada de realizar la función de "main".

Utilizo un "do-while " para mostrar el menú y espero a que el usuario introduzca la opción, que si no fuera valida, devuelve un "print" de "option not selectable". El bucle seguirá funcionando hasta que el usuario introduzca la opción cero.

Cada una de las opciones pasan por un "switch" el cual se encarga de llamar al método correcto. Esta encapsulado en el "try-catch" para que si en el programa ocurre cualquier error, sea recogido aquí.

Introduccion a "AddPlayers()"

En el caso de introducir la opción uno, llamará al método addPlayers de la clase TournamentPlayers, que es la encargada con un bucle de crear y almacenar la información de los jugadores en la arraylist "arPlayersChess" como objetos tipo "PlayersChess".

El objeto tipo playerchess almacena la Id, el nombre y el país como "strings", mientras que las puntuaciones sera almacenada en un "array" tipo "float".

El usuario al introducir cero en el campo id, la aplicación llamará a las funciones "savePlayersToCSV()" y a la funcion "savePlayersToXML()" para almacenar los jugadores tanto en el CSV como en el XML, después volverá al "Main"

Introducción a “savePlayersToCSV()”

El metodo “savePlayersToCSV()” itera todos los objetos jugadores y los almacena como un “string” con saltos de linea entre cada jugador, utilizando el metodo “.toString()” de la clase “PlayersChess” que tiene el formato “id, name, country, score1, score2 and score3” separado por comas.

Después es almacenado en una archivo de texto plano con el nombre de “ChessPlayers.csv” pasado por argumento desde el main.

Introduccion a “savePlayersToXML”

El metodo “savePlayersToXML” se encarga de guardar la información en el archivo “ChessPlayers.xml” pasado por argumento desde el main.

Utiliza “DocumentBuilderFactory” para obtener “DocumentBuilder. Despues, se crea el documento XML vacío con el nombre de raíz “Players”.

Con “for-each” recorre la lista y se genera un nodo por cada jugador.

Dentro de cada nodo de los diferentes jugadores se crean nodos de “id, name, country” y para cada una de las puntuaciones. Ademas se calcula la puntuación total y se almacena en otro nodo.

Por ultimo se transforma y se guarda en el archivo “ChessPlayers.xml”.

Introducción readPlayersFromXML()

Si en el main, el usuario introduce la opción dos, el programa llama a la función “readPlayersFromXML()”. Lee el archivo, normaliza, crea una lista de nodos llamados “nlstPlayer”e itera los nodos “players” e imprime por pantalla su información.

Después vuelve al main.

Introducción convertXMLToHTMLWithXSL()

Si en el main, el usuario introduce la opción tres, el programa llama a la función “convertXMLToHTMLWithXSL”. Toma el XML creado antes y un XSL modelo, se aplica la transformación del “XSLT” al “XML” y lo almacena el resultado en el StringWriter.

Por ultimo escribe el contenido en un archivo HTML utilizando el FileWriter e imprime por terminal “HTML generated successfully” y vuelve al “main”.

Introducción modifyRating()

Si en el main, el usuario introduce la opcion 4, el programa llama a la función "modifyRating". Pide al usuario que introduzca el ID del jugador, la partida y la puntuación que se quiere modificar. El programa busca el elemento del jugador en el XML que coincide con el ID introducido por el usuario y se actualiza la puntuación dejando el resto con los mismos valores.

Transformara el documento y sobrescribe el XML.

También lee el archivo CSV, actualiza la puntuación y sobrescribe el archivo.

Después vuelve al main.

Introducción chessMasterCandidates()

Si en el main, el usuario introduce la opción 5, el programa llama a la funcion "chessMasterCandidates(). Lee el archivo CSV, y jugador por jugador verifica si la suma de sus puntos es igual a tres. Si lo es lo añade a la lista de candidatos.

Guarda los jugadores candidatos en "Candidates.xml" y mediante la función "convertXMLToHTMLWithXSL()" vista antes, genera el HTML

vale, la explicación de esta app termina aquí.

Segunda App

La segunda app consiste en lo mismo pero no usa CSV ni XML. Usa SQLite y MongoDB para almacenar la información de los jugadores de ajedrez.

Main()

El “main” se comporta igual que en la app anterior pero cambiando los metodos a los que llama. Ademas de contener la configuraciones de Mongo y de SQLite.

Para cada tipo de base de datos he realizado una clase diferente para poder gestionarlo desde la clase TournamentPlayers.

PlayersChess()

La clase “PlayersChess()” sigue siendo la misma.

SQLiteOperations()

La clase “SQLiteOperations()” contiene los métodos para conectar, crear tablas si no existieran, imprimir tabla, borrar datos y cerra conexión con la base de datos de SQLite. Que se irán llamando según la necesidad desde el “main” o desde “tournamentplayers”

MongoOperations()

La clase “MongoOperations()” contiene los métodos para conectar, crear colecciones si no existieran, borrar colecciones si existieran, verificar colecciones y cerrar conexiones. Que se llamarán desde el “main” o desde “tournamentplayers”

TournamentPlayers()

La clase “TournamentPlayers() sirve para todo lo relacionado con jugadores de ajedrez en el torneo.

El metodo getChessPlayersAndScore() es la misma que AddPlayers en la otra aplicación.

El metodo “writePlayersToDB_usingJDBC” escribe la información almacenada en el array PlayersChess y la vuelca sobre la base de datos SQLite utilizando JDBC.

El metodo “readPlayersToDB_usingJDBC” lee la información de la base de datos SQLite utilizando JDBC y la devuelve al array “PlayersChess”

El metodo “dumpDataFromSQLiteToMongoDB” importa la información de los jugadores desde la base de datos SQLite a la base de datos MongoDB

El metodo “writePlayersToDB_usingMongoDB” coge la información del ArrayList y la introduce en la base de datos de MongoDB.

El metodo “listAllChessPlayersFromMongoDB” imprime por consola la informacion de todos los jugadores almacenados en la base de datos de MongoDB.