The solution I've proposed consists of a main method in which I initialize an instance of ChessTournament, I display the menu with the options and ask the user to choose one of them (depending on the option, I execute different functions) and finally, if the option is 0, I terminate the program, un método ChessPlayer que permite crear instancias de objetos del mismo nombre (cada ChessPlayer tiene un identificador, un nombre, un país, 3 puntuaciones y el total, que es la suma de las 3 anteriores) and, finally, a ChessTournament method that contains an ArrayList of ChessPlayer where all the chess tournament players are collected, and it provides access to all the functions that the application can perform (save the data of all the players to an XML and CSV file, show the data of all the players on console, modify the score of one of the games for a player, and as a result, update their total score in both the XML and CSV files, generate an HTML file that displays all player data and another HTML file that shows only the data of players with a total score of 3). I've also used a class called IOTools to receive user data, to which I've added two functions: one for receiving strings without spaces and another to prompt the user to press a key before continuing when one of the functions ends.

The most significant difficulty I encountered when creating the application was, on the one hand, remembering the numerous steps required to follow the DOM protocol, and on the other hand, when it came to modifying the CSV file because I had never had to write and read from the same document. Therefore, I didn't know that I needed to close the BufferedReader before creating the BufferedWriter, which led to a period of research and a trial-and-error process.

One way to expand the program could be , since you create an instance of ChessTournament, create an instance of ChessTournament for each tournament, where you ask the user for the tournament's name and generate a custom-named XML and CSV file for each tournament, along with allowing the user to specify the directory where they want to save the files (instead of a constant path). Adding functions to modify all player data except for the ID, such as their name and country, would indeed be another valuable expansion of the application. This would allow users to update player information as needed, making the application more versatile and user-friendly.