Case Study-Tennis player Game

Part 1 : Write code and Test-cases

There is this class called TennisPlayer with the following attributes:

1. playerCode of type int

2. playerName of type String

3. careerDetail - of type PlayerCareer defined below:

PlayerCareer has:

1. matchesPlayed - int

2. matchesWon - int

3. currentRanking - int

4. bestRanking - int

5. preferredCourt - String - can have only 1 of 4 values [Grass, Synthetic, Clay, Others] per entry.

6. lastFought - int - playerCode of another player against whom this player played.

Code for the following:

1. Create at least 10 tennis players as a list with valid values for all attributes.

2. Write a method playMatch(int playerCode) in TennisPlayer. The method should return an int, whose values can be:

1 - if the player wins, -1 if the player loses. currentRanking decreases by 1 on a win and increases by 1 on a loss. Either way, this method should update the PlayerCareer for both the players for the relevant fields.

*Note: use **Random** method to decide the winner between two players*

3. Create your own data structure to store the players based on playerCode.

4. Write a method deletePlayer(int playerCode) which finds and deletes that player in the list created in #3. Write your own Exception, "PlayerNotFoundException". The method should throw this exception if no player is found with that playerCode in the list.

5. Create another collection that stores all the TennisPlayers based on currentRanking in ascending order. If current ranking is the same, then sort by playerName in ascending order [hint: use your own comparator]. Print the sorted list.

Please use correct access modifiers and write code for getters, setters and other methods as required. Write separate test methods for each of the items in the list above.