Comments / Answers

1. Program requests to enter a number of players in tournament, why? How I can know this value? Why I should enter it anyway? All data exists in CSV files, you can count it.
This issue was resolved. The program counts files on its own.

2. Adding a new sport is complicated, need to change many places.
Now when you want to add a new game to the tournament you have to implement the interface "Game" and create one more if-else statement in the "Main" method.

3. Program always assumes we have 2 teams A and B, why? this can change at any point of time, code should be flexible and handle any teams from files.
Now you can use any names of team.

4. Bad utilization of SOLID principles, example : game implementations does to many things like file parsing and business logic of counting scores (Handball for reference).
The methods of the program were rewritten according to SOLID principles.

5. All Game implementations are almost identical, Handball and BasketBall does the same thing with small difference, why not extract common logic to single place?
The common logic was put on the default method of the interface. (Methods "createTeam", "countRatingPoints" and "defineWinnerTeam" have different logic so that each new sport must override them.

6. Player object has field for all type of players, for all games. This is bad design as in case of new sport we will add more fields there and that object will be huge. The proper way is to have only common field in Player and extend it for each type of Sport which will contain sport specific fields.
The class Player is extended by Basketball and Handball classes.

7. Why Game is abstract class? It has only 2 abstract methods without any internal logic, its a classic interface.
Now the program uses the interface "Game".

Description of the program

1. The program creates the object of the tournament which will be used farther in the program.

2. The program checks files in the “CSV_files” folder for the wrong format of the files.

3. The program checks for the uniqueness of the nicknames in each game.

4. The program defines the type of the game currently it can be handball or basketball.

5. If the first row of the CSV file contains word “basketball” the program will create an object of basketball. Next, the program uses the object “basketball” and calls the method “createTeam” and passes a CSV file to this method. The method returns a list of objects which will be assigned to the list “createdObjects”.

6. The program uses the object “basketball” and calls the method “countRatingPoints” and passes the list “createdObjects” to this method. The method counts the rating points of each player in the list.

7. The program uses the object “basketball” and calls the method “defineWinnerTeam” and passes to this method the list “createdObjects”. The method defines the winning team and returns the list of players of the winning team which is assigned to the list “winnerTeam”.

8. The program uses the object “basketball” and calls the method “add10PointsToWinnerTeam” and passes the list “winnerTeam”. The method adds 10 points to each player of the winning team.

9. Now the program uses the object “tournament” and calls the method “addToRatingList” and passes the list “createdObjects” to this method. If the map "raitingList" is empty method puts the nickname and rating points of each player into the map. if map "raitingList" is not empty the method adds rating points to each existing nickname in the map. If the map does not contain the nickname the method adds a new nickname and rating point into the map.

10. The program uses the object “tournament” and calls the method “displayToConsole”. The method displays the map “ratingList” to the console.

11. The program uses the object “tournament” and calls the method “defineMVP”. The method displays the nickname and rating points of the most valuable player of the tournament.

** If the program defines the word “handball” on the fifth step. All methods which were used to “basketball” object will be used to “handball” object.*

Diagram of the program