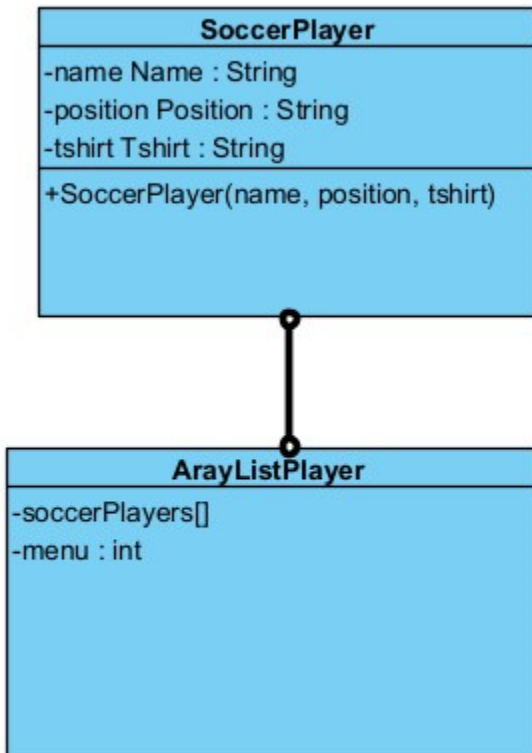


5) Code Quality 6/10 pts.
TOTAL: 40 pts.

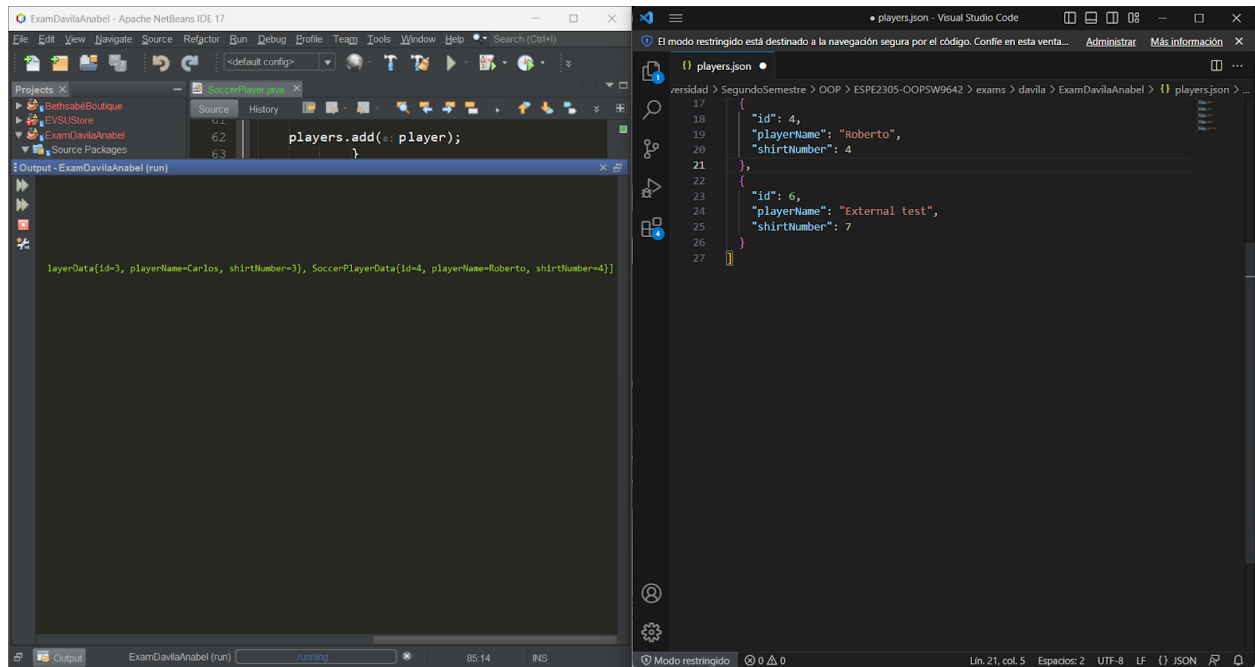
Evidence

1) class modeling (Class diagram)



2) Running program (cmd line executing the program)

```
Select an option:
1. Add a player
2. Print the data base of the players
3. Print all the players
0. Out
```



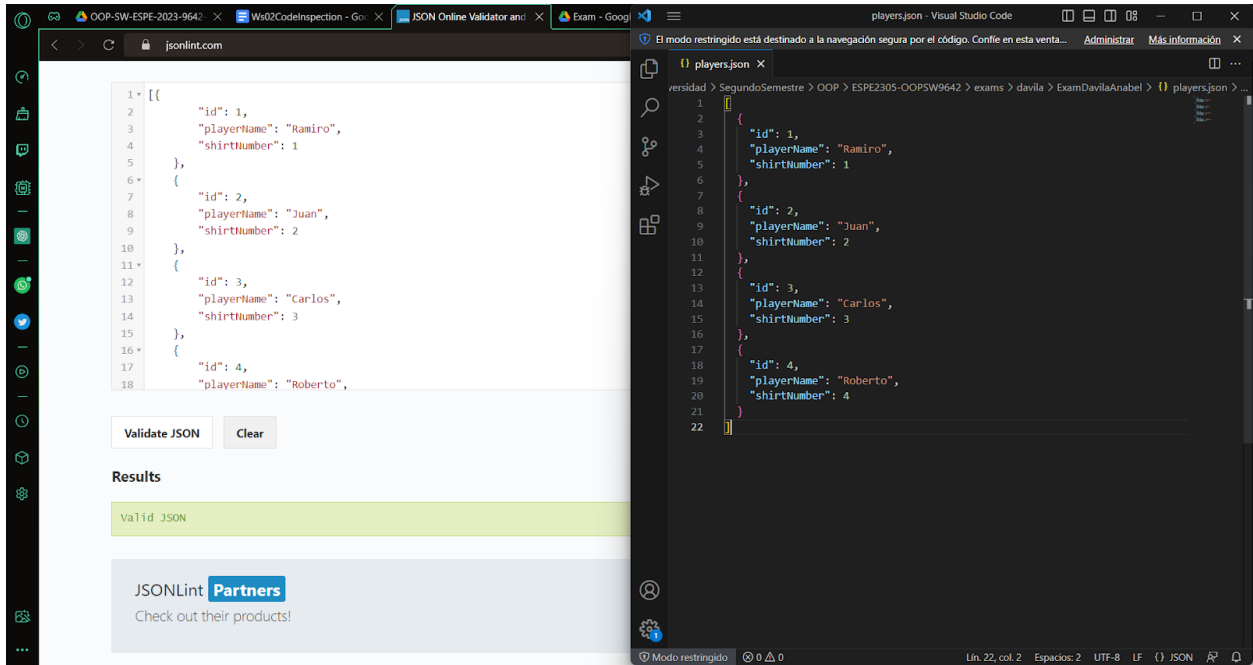
Default creations and no modify.

3) Printing/counting/deleting objects from JSON File (according to the roster number)

```
4
Add the name of the player
Test4
Add the shirt number of the player
4
  Select an option:
1. Add a player
2. Print the data base of the players
3. Print all the players
0. Out
1
Add the id of the player
5
Add the name of the player
Test5
Add the shirt number of the player
5
  Select an option:
1. Add a player
2. Print the data base of the players
3. Print all the players
0. Out
2
Data base file created
  Select an option:
1. Add a player
2. Print the data base of the players
3. Print all the players
0. Out
3
```

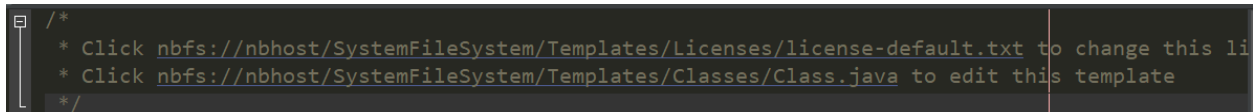
```
[{"id": 2, "playerName": "Test2", "shirtNumber": 2}, {"id": 3, "playerName": "Test3", "shirtNumber": 3}, {"id": 4, "playerName": "Test4", "shirtNumber": 4}, {"id": 5, "playerName": "Test5", "shirtNumber": 5}]
```

4) Saving JSON data (screenshot of the json file)

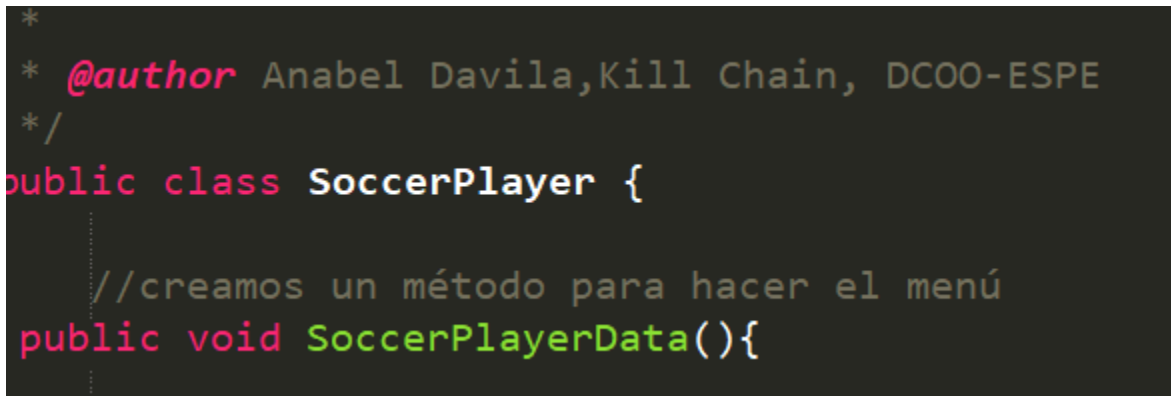


5) Code Quality

(lines of code with stinky code)



Unnecessary headers



Unnecessary comments

```

public class SoccerPlayer {

    //creamos un método para hacer el menú
    public void SoccerPlayerData(){

        SoccerPlayerData player;
        ArrayList<SoccerPlayerData> players = new ArrayList<>();

        Scanner collectData = new Scanner(source: System.in);

        player = new SoccerPlayerData (id: 1, playerName: "Ramiro", shirtNumber: 1);
        players.add(e: player);
        player = new SoccerPlayerData (id: 2, playerName: "Juan", shirtNumber: 2);
        players.add(e: player);
        player = new SoccerPlayerData (id: 3, playerName: "Carlos", shirtNumber: 3);
        players.add(e: player);
        player = new SoccerPlayerData (id: 4, playerName: "Roberto", shirtNumber: 4);
        players.add(e: player);

        int menu;

        do{

            System.out.println(x: " Select an option: ");
            System.out.println(x: "1. Add a player ");
            System.out.println(x: "2. Print the data base of the players");
            System.out.println(x: "3. Print all the players");

```

Menu options in the SoccerPlayer class, instead of main.

```

do{

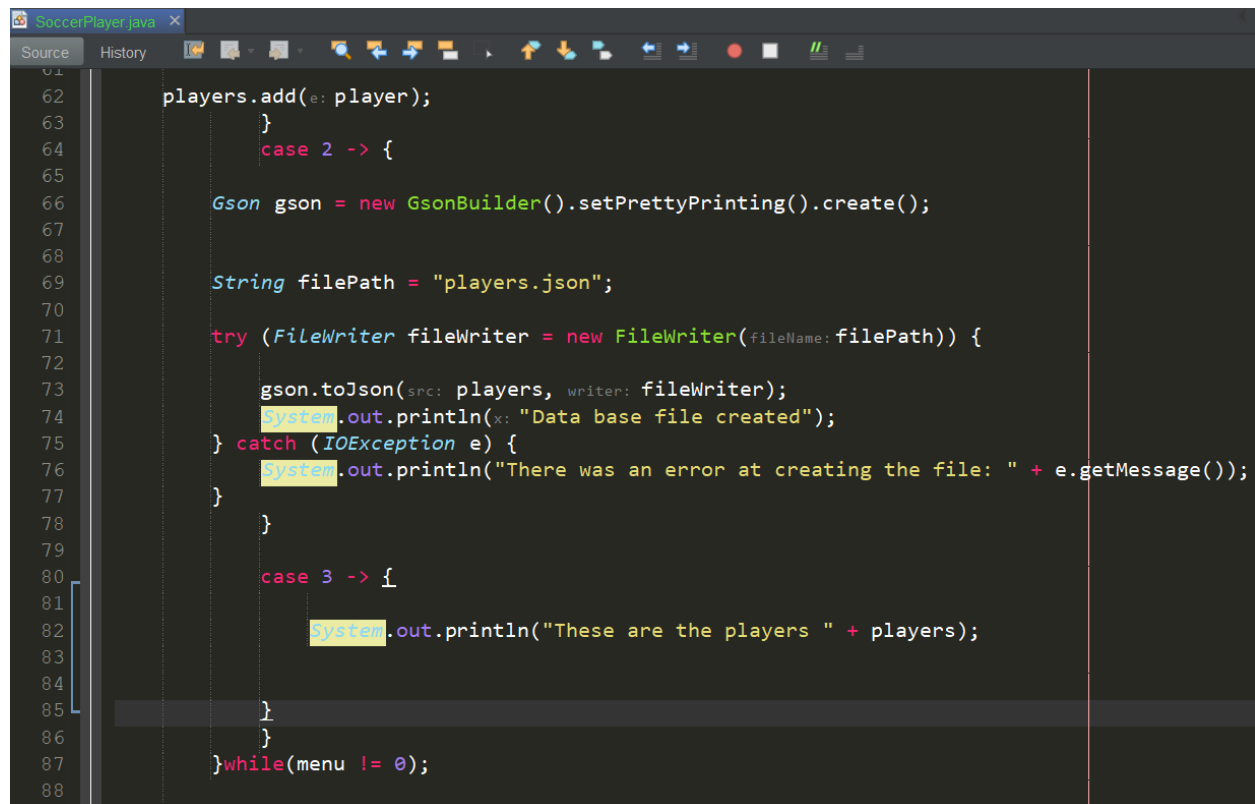
    System.out.println(x: " Select an option: ");
    System.out.println(x: "1. Add a player ");
    System.out.println(x: "2. Print the data base of the players");
    System.out.println(x: "3. Print all the players");
    System.out.println(x: "0. Out");

    menu = collectData.nextInt();
    switch (menu){
    case 1 -> {
        System.out.println(x: "Add the id of the player");
        int id = collectData.nextInt();
        System.out.println(x: "Add the name of the player");
        String playerName = collectData.next();
        System.out.println(x: "Add the shirt number of the player");
        int shirtNumber = collectData.nextInt();
        player = new SoccerPlayerData (id, playerName, shirtNumber);

        players.add(e: player);
    }
}
}

```

Bad Indentation.



```

SoccerPlayer.java
Source History
61
62     players.add(e: player);
63     }
64     case 2 -> {
65
66         Gson gson = new GsonBuilder().setPrettyPrinting().create();
67
68         String filePath = "players.json";
69
70         try (FileWriter fileWriter = new FileWriter(fileName: filePath)) {
71
72             gson.toJson(src: players, writer: fileWriter);
73             System.out.println(x: "Data base file created");
74         } catch (IOException e) {
75             System.out.println("There was an error at creating the file: " + e.getMessage());
76         }
77     }
78
79
80     case 3 -> {
81
82         System.out.println("These are the players " + players);
83
84     }
85 }
86 }
87 }while(menu != 0);
88

```

JSON File CRUD in the same SoccerPlayer class.

Inspector: Joan Cobeña