Solution Review: Aggregate `Em All!

This is the solution to the exercise, "Aggregate `Em All!" with an explanation.

WE'LL COVER THE FOLLOWING

Solution

```
// Player class
class Player {

  String name;
  int id;
  String team;

  Player(String name, int id, String team) {
    this.name = name;
    this.id = id;
    this.team = team;
  }

}

/* Team class contains a list of Player
Objects.*/
class Team {

  String name;
  private List<Player> players;

  Team(String name, List<Player> players) {
    this.name = name;
    this.players = players;
  }

  public List<Player> getPlayers() { // This function returns the "players"
    return players;
  }

}

/* School class contains a list of Team
Objects.*/
```

```
class School {

  String schoolName;
  private List<Team> teams;

  School(String schoolName, List<Team> teams) {
    this.schoolName = schoolName;
    this.teams = teams;
  }

  /* Count total players of all teams
    in a given school */
  public int getTotalPlayersInSchool() {
    int noOfPlayers = 0;
    List<Player> players;

    for(Team team : teams) {
      players = team.getPlayers();
      for(Player p : players) {
        noOfPlayers++;
      }
    }
    return noOfPlayers;
  }

}

// Main class
class Main {

  public static void main (String[] args) {
    /* Declaring all the players */
    Player p1 = new Player("Harris", 1, "Red");
    Player p2 = new Player("Carol", 2, "Red");
    Player p3 = new Player("Johnny", 1, "Blue");
    Player p4 = new Player("Sarah", 2, "Blue");

    /* Making a List of
        "Red" team Players. */
    List <Player> red_players = new ArrayList<Player>();
    red_players.add(p1);
    red_players.add(p2);

    /* Making a List of
        "Blue" team Players. */
    List <Player> blue_players = new ArrayList<Player>();
    blue_players.add(p3);
    blue_players.add(p4);

    /* Declaring Team objects */
    Team red = new Team("Red", red_players);
    Team blue = new Team("Blue", blue_players);

    // Creating a list of teams and adding "red" and "blue" teams to it.
    List <Team> teams = new ArrayList<Team>();
    teams.add(red);
    teams.add(blue);

    // Creating an instance of School.
    School mySchool = new School("ABC", teams);

    System.out.println("Total players in my school: ");
    // Getting total prayers in the school.
    System.out.println(mySchool.getTotalPlayersInSchool());
  }

}
```

Explanation

- Line **21**: `Team` contains a list of `Player` objects, which are then initialized in the constructor.
- Line **28**: The function `getPlayers` returns the list of **players** in a “Team” object.
- Line **39**: `School` contains a list of different `Team` objects.
- Line **48**: The function, “`getTotalPlayersInSchool()`” goes through all the teams in a school and then again loops over all the **players** in a **team** instance to count all the players in the **school**.