# 03. Basketball Players

*One of your best friends is the manager of the National Amateur Basketball League. He asked you to create a program, which will help him manage the players and their teams.*

## Preparation

Download the skeleton provided in Judge. **Do not** change the **StartUp** class or its **namespace**.

**Note**: The target framework of your project must be .NET Core 3.1.

## Problem description

Your task is to create a catalog of teams, containing information for various basketball players.

# Player

You are given a class **Player,** create the following properties:

* **Name: string**
* **Position: string**
* **Rating: double**
* **Games: int**
* **Retired: boolean** – **false by default**

The class **constructor** should receive **(name, position, rating, games)**.

The class should also have a method:

* Override the **ToString()** method in the format:

**"-Player: {name}**

**--Position: {position}**

**--Rating: {rating}**

**--Games played: {games}"**

# Team

Next, a class named **Team**is given, that has a **collection**(**players**) of type **Player**. All the entities of the **Players** collection have the **same** properties. The **Team** has also some additional properties:

* **Name: string**
* **OpenPositions: int**
* **Group: char**

The **constructor** of the **Team** class should receive the **name, openPositions** and **group.**

Implement the following features:

* Getter **Count** - returns the count of the players in the team.
* string AddPlayer(Player player) – **adds** a player to the team's collection, **if** **there are open positions**. Before adding a player, check:
  + - * + If the **name** or **position** is **null or empty**, return **"Invalid player's information.".**
  + If there are no more open positions, return "**There are no more open positions.".**
  + If the **rating** is **under 80**,return **"Invalid player's rating.".**
  + Otherwise, return: **"Successfully added {playerName} to the team. Remaining open positions: {openPositions}."** and decrease the **OpenPositions** property of the team.
* bool RemovePlayer(string name) – removes a player by **given name.**
  + - If such **exists,** returns **true**;
    - Otherwise, returns **false.**

**Note: Remember to update the OpenPositions property!**

* int RemovePlayerByPosition(string position) – removes **all players** by the given **position.**
  + - If such **exist(s), returns the count of the removed players;**
    - Otherwise, returns **0.**

**Note: Remember to update the OpenPositions property!**

* Player RetirePlayer(string name) method – **retire** (**set** his Retired **property** to **true,** without removing him from the collection) the **player** with the **given name, if he exists**. As a result, **return** the **player, or null, if he don't exist.**
* List<Player> AwardPlayers (int games) method – **return** a **list** with **all players** that have participated in **games** games or more.
* **Report()** –returns a string with information about the team and players who are **not retired** in the following format:

**"Active p**layers **competing for Team {team} from Group {group}:  
{**Player**1}  
{**Player**2}  
{…}**"

**Note: Do not use** "\n\r" **for a new line.**

# Constraints

* The **names** of the players will be **always unique**.
* You will always have a player added before receiving methods, manipulating the teams' players.

# Examples

This is an example of how the **Team class** is **intended to be used**.

|  |
| --- |
| Sample code usage |
| // Initialize the repository (Team)  Team team = new Team("BHTC", 5, 'A');  // Initialize entity  Player firstPlayer = new Player("Viktor", "Center", 97.5, 10);  // Print player  Console.WriteLine(firstPlayer);  /\*  -Player: Viktor  --Position: Center  --Rating: 97.5  --Games played: 10  \*/  // Add Player  Console.WriteLine(team.AddPlayer(firstPlayer));  /\*  Successfully added Viktor to the team. Remaining open positions: 4.  \*/  // Check count of added players  Console.WriteLine(team.Count);  /\*  1  \*/  // Remove Player  Console.WriteLine(team.RemovePlayer("Slavi"));  /\*  False  \*/  Player secondPlayer = new Player ("Slavi", "Point Guard", 94.3, 47);  Player thirdPlayer = new Player ("Evgeni", "Shooting Guard", 93.7, 16);  Player fourthPlayer = new Player ("Momchil", "Small forward", 67.9, 3);  Player fifthPlayer = new Player ("Vasil", "Power forward", 86.9, 10);  Player sixthPlayer = new Player ("Stefan", "Center", 95.6, 25);  Player seventhPlayer = new Player ("Ivan", " Small forward ", 98.5, 89);  // Add players  Console.WriteLine(team.AddPlayer(secondPlayer));  Console.WriteLine(team.AddPlayer(thirdPlayer));  Console.WriteLine(team.AddPlayer(fourthPlayer));  Console.WriteLine(team.AddPlayer(fifthPlayer));  Console.WriteLine(team.AddPlayer(sixthPlayer));  Console.WriteLine(team.AddPlayer(seventhPlayer));  /\*  Successfully added Slavi to the team. Remaining open positions: 3  Successfully added Evgeni to the team. Remaining open positions: 2  Invalid player's rating  Successfully added Vasil to the team. Remaining open positions: 1  Successfully added Stefan to the team. Remaining open positions: 0  There are no more open positions.  \*/  // Retire player by name  Console.WriteLine(team.RetirePlayer("Slavi"));  /\*  -Player: Slavi  --Position: Point Guard  --Rating: 94.3  --Games played: 47  \*/  // Award players  List<Player> players = team.AwardPlayers(15);  foreach(var playerToBeAwarded in players)  {  Console.WriteLine(playerToBeAwarded);  }  /\*  -Player: Slavi  --Position: Point Guard  --Rating: 94.3  --Games played: 47  -Player: Evgeni  --Position: Shooting Guard  --Rating: 93.7  --Games played: 16  -Player: Stefan  --Position: Center  --Rating: 95.6  --Games played: 25  \*/  // Remove player by position  Console.WriteLine(team.RemovePlayerByPosition("Center"));  /\*  2  \*/  // Report  Console.WriteLine("----------------------Report----------------------");  Console.WriteLine(team.Report());  /\*  Active players competing for Team BHTC from Group A:  -Player: Evgeni  --Position: Shooting Guard  --Rating: 93.7  --Games played: 16  -Player: Vasil  --Position: Power forward  --Rating: 86.9  --Games played: 10  \*/ |

# Submission

Zip all the files in the project folder except **bin** and **obj** folders.