Fighting Arena

Preparation

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

Problem description

Your task is to create an arena which stores gladiators by creating the classes described below.

First, write a C# class **Weapon** with the following properties:

- **Size: int**
- **Solidity: int**
- **Sharpness: int**

The class **constructor** should receive **size, solidity and sharpness**.

Next, write a C# class **Stat** with the following properties:

- **Strength: int**
- **Flexibility: int**
- **Agility: int**
- **Skills: int**
- **Intelligence: int**

The class **constructor** should receive **strength, flexibility, agility, skills and intelligence.**

Next, write a C# class **Gladiator** with the following properties and methods:

- **Name: string**
- **Stat: Stat**
- **Weapon: Weapon**
- **GetTotalPower(): int –** return the sum of the stat properties plus the sum of the weapon properties.
- **GetWeaponPower(): int -** return the sum of the weapon properties.
- **GetStatPower(): int -** return the sum of the stat properties.

The class **constructor** should receive **name, stat and weapon** and **override ToString()** in the following format:

```
"[Gladiator name] - [Gladiator total power]"
"  Weapon Power: [Gladiator weapon power]"
"  Stat Power: [Gladiator stat power]"
```

Write a C# class **Arena** that has **gladiators** (a collection which stores the class **Gladiator**).

```csharp
public class Arena
{
    // TODO: implement this class
}
```

Follow us:

The class **constructor** should initialize the **gladiators** with a new instance of the collection**.** Implement the following features:

- Field **gladiators** – **collection** that holds added gladiators
- **Property Name - string**
- Method **Add(Gladiator gladiator)** – adds an gladiator to the arena.
- Method **Remove(string name)** – removes an gladiator by given name.
- Method **GetGladitorWithHighestStatPower()** – returns the Gladiator which has the highest stat.
- Method **GetGladitorWithHighestWeaponPower()** – returns the Gladiator which poses the weapon with the highest power.
- Method **GetGladitorWithHighestTotalPower()** – returns the Gladiator which has the highest total power.
- Getter **Count** – returns the number of stored heroes.
- Override **ToString()** – by the format below.
  "[Arena name] - [count of gladiators] gladiators are participating."

Constraints

- The names of the gladiators will be always unique.
- The weapons and the stat properties of the gladiators will always be with positive values.
- The weapon power, stat power and total power of the gladiators will always be different.
- You will always have a gladiator with the highest stat, weapon and total power.

Examples

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

<table>
<tr><th>Sample code usage</th></tr>
<tr><td>

```
//Creates arena
Arena arena = new Arena("Armeec");

//Creates stats
Stat firstGlariatorStat = new Stat(20, 25, 35, 14, 48);
Stat secondGlariatorStat = new Stat(40, 40, 40, 40, 40);
Stat thirdGlariatorStat = new Stat(20, 25, 35, 14, 48);

//Creates weapons
Weapon firstGlariatorWeapon = new Weapon(5, 28, 100);
Weapon secondGlariatorWeapon = new Weapon(5, 28, 100);
Weapon thirdGlariatorWeapon = new Weapon(50, 50, 50);

//Creates gladiators
Gladiator firstGladiator = new Gladiator("Stoyan", firstGlariatorStat, firstGlariatorWeapon);
Gladiator secondGladiator = new Gladiator("Pesho", secondGlariatorStat, secondGlariatorWeapon);
Gladiator thirdGladiator = new Gladiator("Gosho", thirdGlariatorStat, thirdGlariatorWeapon);

//Adds gladiators to arena
arena.Add(firstGladiator);
arena.Add(secondGladiator);
arena.Add(thirdGladiator);

//Prints gladiators count at the arena
Console.WriteLine(arena.Count);

//Gets strongest gladiator and print him
Gladiator strongestGladiator = arena.GetGladitorWithHighestTotalPower();
Console.WriteLine(strongestGladiator);
```

</td></tr>
</table>

```csharp
//Gets gladiator with the strongest weapon and print him
Gladiator bestWeaponGladiator = arena.GetGladitorWithHighestWeaponPower();
Console.WriteLine(bestWeaponGladiator);

//Gets gladiator with the strongest stat and print him
Gladiator bestStatGladiator = arena.GetGladitorWithHighestStatPower();
Console.WriteLine(bestStatGladiator);

//Removes gladiator
arena.Remove("Gosho");

//Prints gladiators count at the arena
Console.WriteLine(arena.Count);

//Prints the arena
Console.WriteLine(arena);
```

Submission

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