

## Упражнение 4 – Допълнителни задачи

### Problem 1. Първи и резервен Отбор

Създайте клас Team. Добавете към него всички обекти person, които въвеждате. Обектите person, по-млади от 40 отиват в първи отбор, другите - в резервен. На края изведете броя на участниците в първия и резервния отбор.

Класът трябва да има **private** полета за:

- name: string
- firstTeam: List<Person>
- reserveTeam: List<Person>

Класът трябва да има **constructors**:

- Team(string name)

Трябва да има също **public** методи за:

- AddPlayer(Person person): void
- FirstTeam: IReadOnlyCollection
- ReserveTeam: IReadOnlyCollection

Трябва да можете да използвате класа:

```
private string name;
private List<Person> firstTeam;
private List<Person> reserveTeam;

public Team(string name)
{
    this.name = name;
    this.firstTeam = new List<Person>();
    this.reserveTeam = new List<Person>();
}
```

**НЕ бива** да използвате клас като този:

```
Team team = new Team("Gorno Nanadolnishte");
foreach (var player in persons)
{
    if (player.Age < 40)
        team.FirstTeam.Add(player);
    else
        team.ReserveTeam.Add(player);
}
```



### Примери

Вход	Изход
5 Asen Ivanov 20 2200 Boiko Borisov 57 3333 Ventsislav Ivanov 27 600 Grigor Dimitrov 25 666.66 Boiko Angelov 35 555	First team have 4 players Reserve team have 1 players

## Решение

Добавете нов клас Team. Неговите полета и конструктор да бъдат например такива:

```
private string name;
private List<Person> firstTeam;
private List<Person> reserveTeam;

public Team(string name)
{
    this.name = name;
}
```

Свойствата за FirstTeam and ReserveTeam имат само getters:

```
public IReadOnlyCollection<Person> FirstTeam
{
    get { return this.firstTeam.AsReadOnly(); }
}

public IReadOnlyCollection<Person> ReserveTeam
{
    get { return this.reserveTeam.AsReadOnly(); }
}
```

Ще има само един метод, който добавя играчи към отбора

```
public void AddPlayer(Person player)
{
    if (player.Age < 40)
    {
        firstTeam.Add(player);
    }
    else
    {
        reserveTeam.Add(player);
    }
}
```

## Problem 2. \*\*Създаване на футболен отбор

Футболен отбор има променлив **брой играчи, име и рейтинг**. Един играч има **име** и **статистика**, които са в основата на неговото ниво на умения. Един играч има статистика са **издръжливост, Спринт, дрибъл, подавания и стрелба**. Всяка статистика може да бъде в диапазона [0..100]. Общото ниво на умение на играч се изчислява като средна стойност на статистиките си. Само името на играча и неговата статистика трябва да бъдат видими за всички от външния свят. Всичко останали данни трябва да бъдат скрити. Отборът трябва да показва име, рейтинг (изчислена от нивата на средните умения на всички играчи в отбора и закръглена до цяло число) и методи за добавяне и премахване на играчи. Вашата задача е да моделирате екипа и играчите, чрез правилното използване на принципите на капсулиране. Покажи само свойствата, които трябва да бъдат видими и валидирайте данните по подходящ начин.

## Валидация на данните

- Името не трябва да е празно, null, empty или да е само от интервали. Иначе, изведете "A name should not be empty. "
- Stats трябва да е в обхвата 0..100. Иначе изведете "[Име на статистиката] should be between 0 and 100. "
- Ако получите команда да премахнете липсващ играч, изведете "Player [име на играча] is not in [Име на отбора] team. "
- Ако получите команда да добавите играч към липсващ отбор, изведете "Team [име на отбор] does not exists."
- Ако получите команда за показване на stats за липсващ отбор, изведете "Team [име на отбор] does not exists."

## Примери

Вход	Изход
Team;Arsenal Add;Arsenal;Kieran_Gibbs;75;85;84;92;67 Add;Arsenal;Aaron_Ramsey;95;82;82;89;68 Remove;Arsenal;Aaron_Ramsey Rating;Arsenal END	Arsenal - 81
Team;Arsenal Add;Arsenal;Kieran_Gibbs;75;85;84;92;67 Add;Arsenal;Aaron_Ramsey;195;82;82;89;68 Remove;Arsenal;Aaron_Ramsey Rating;Arsenal END	Endurance should be between 0 and 100. Player Aaron_Ramsey is not in Arsenal team. Arsenal - 81
Team;Arsenal Rating;Arsenal END	Arsenal - 0

## Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма **"Обучение за ИТ кариера"** на МОН за подготовка по професия "Приложен програмист".



- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

