

# C# OOP - July 2017 - Olympics System

---

## General Description

Implement an olympian tracking system for the next Olympic games. The application already accepts commands and outputs text for each submitted command, you just need to write the OOP behind most of the command. You can create boxers and sprinters, as well as listing them. Make sure to follow all the Object Orientated Programming practices and conventions that we have talked about during the lectures and don't let the length of this description intimidate you, read it carefully and start hacking!

## Classes

Read the description **carefully** before you proceed to the code base. Read the code base **carefully** before you start implementing the down-mentioned requirements.

Implement and validate the following classes:

- **Boxer** has:
  - **FirstName** that has length in the interval of [2,20] symbols.
  - **LastName** that has length in the interval of [2,20] symbols.
  - **Country** that has length in the interval of [3,25] symbols.
  - **Category** that can be either **Flyweight**, **Featherweight**, **Lightweight**, **Middleweight**, **Heavyweight**.
  - **Wins** that is in the interval of [0,100].
  - **Losses** that is in the interval of [0,100].
- **Sprinter** has:
  - **FirstName** has length in the interval of [2,20] symbols.
  - **LastName** has length in the interval of [2,20] symbols.
  - **Country** has length in the interval of [3,25] symbols.
  - **PersonalRecords** that are in the format of key/value pair.
- **OlympicsFactory** has:
  - **CreateBoxer(...)** that needs to be implemented.
  - **CreateSprinter(...)** that needs to be implemented.

# Commands

You are given a set of commands, some of which you need to implement yourself.

- **createboxer** [firstname] [lastname] [country] [category] [wins] [losses] - Creates a new **Boxer** with the following parameters.
- **createsprinter** [firstname] [lastname] [country] [records]- Creates a new **Sprinter** with the following parameters. The records are in the following format [range]/[time].
- **listolympians** [key] [order] - Lists the olympians, sorted by a certain **key** in a certain ordering. All parameters are optional and should have default value. The keys can be **firstname**, **lastname**, **country**. The order can be either **asc** or **desc**. The available keys are **firstname**, **lastname**, **country**.
  - When no parameters are passed, the default key is **firstname**, and the default order is **ascending** (ex: listolympians)
  - When only key is passed the default order is ascending (ex: listolympians lastname)
- **end** - Terminates the application. (Hint: Already implemented. Thank me later.)

**All commands are case insensitive, except their parameters!** Each command is represented in the code base as a separate class, that is invoked by the Engine. **Do not change** the command names, as they are used to resolve the passed in the console strings.

## Architecture

Let's talk a bit about how the system works (you are already provided with all of this stuff, there is no need to implement it). There is an **Engine** located in the **Core** namespace that has a loop that cycles until the **end** command is submitted. With each cycle, it takes the input, passes it to the command parser that find the command with that name and executes it with those parameters. All commands are located in the **Core.Commands** namespace. The commands themselves use the **OlympicsFactory** located in the **Core.Factories** namespace to create the needed objects. After the command executes, it returns a result message to the **Engine** that prints it to the console and then the cycle beings again. In the **Engine**, there is a try-catch block that catches every possible exception type and prints the exception's message to the console. Do not bother reading those classes, your focus should be on the **Models** namespace, where you need to place the classes you create, using the provided interfaces in the **Contracts** namespace. The result from the execution of the commands is printed after executing the **end** command.

## Constraints

- You are allowed create new **classes, interfaces, enumerations and namespaces** in the **Olympics** namespace.
- You are allowed to modify the **OlympicsFactory** in the **Core.Factories** namespace.
- You are allowed to modify the **CreateBoxerCommand** **Core.Commands** namespace. **(Do not change the name!)**
- You are allowed to modify the **CreateSprinterCommand** **Core.Commands** namespace. **(Do not change the name!)**
- You are allowed to modify the **ListOlympiansCommand** **Core.Commands** namespace. **(Do not change the name!)**
- **You are NOT allowed to modify any other existing interfaces!**

- **You are NOT allowed to modify any other existing classes, enumerations and namespaces!**

## Example

Test each command separately, after you implement it. When you are done, use the input below to fully test your application.

### Input

```
listolympians
createboxer Wladimir Klitschko Ukraine heavyweight 64 5
createboxer Anthony Joshua GreatBritain heavyweight 19 0
listolympians firstname
createsprinter Usain Bolt Jamaica 100/9.58 200/19.19
listolympians
createsprinter Asaffa Powell
createsprinter Asaffa Powell Jamaica 100/9.72 200/19.90
createsprinter U Bolt Jamaica
createsprinter Usain Bolt Ja
listolympians lastname desc
createsprinter Tyson Gay USA
listolympians
end
```

### Output

```
NO OLYMPIANS ADDED
#####
Created Boxer
BOXER: Wladimir Klitschko from Ukraine
Category: Heavyweight
Wins: 64
Losses: 5
#####
Created Boxer
BOXER: Anthony Joshua from GreatBritain
Category: Heavyweight
Wins: 19
Losses: 0
#####
Sorted by [key: firstname] in [order: asc]
BOXER: Anthony Joshua from GreatBritain
Category: Heavyweight
Wins: 19
Losses: 0
BOXER: Wladimir Klitschko from Ukraine
```

```

Category: Heavyweight
Wins: 64
Losses: 5
#####
Created Sprinter
SPRINTER: Usain Bolt from Jamaica
PERSONAL RECORDS:
100m: 9.58s
200m: 19.19s
#####
Sorted by [key: firstname] in [order: asc]
BOXER: Anthony Joshua from GreatBritain
Category: Heavyweight
Wins: 19
Losses: 0
SPRINTER: Usain Bolt from Jamaica
PERSONAL RECORDS:
100m: 9.58s
200m: 19.19s
BOXER: Wladimir Klitschko from Ukraine
Category: Heavyweight
Wins: 64
Losses: 5
#####
ERROR: Parameters count is not valid!
Created Sprinter
SPRINTER: Asaffa Powell from Jamaica
PERSONAL RECORDS:
100m: 9.72s
200m: 19.9s
#####
ERROR: First name must be between 2 and 20 characters long!
ERROR: Country must be between 3 and 25 characters long!
Sorted by [key: lastname] in [order: desc]
SPRINTER: Asaffa Powell from Jamaica
PERSONAL RECORDS:
100m: 9.72s
200m: 19.9s
BOXER: Wladimir Klitschko from Ukraine
Category: Heavyweight
Wins: 64
Losses: 5
BOXER: Anthony Joshua from GreatBritain
Category: Heavyweight
Wins: 19
Losses: 0
SPRINTER: Usain Bolt from Jamaica
PERSONAL RECORDS:
100m: 9.58s
200m: 19.19s
#####
Created Sprinter

```

```
SPRINTER: Tyson Gay from USA
NO PERSONAL RECORDS SET
#####
Sorted by [key: firstname] in [order: asc]
BOXER: Anthony Joshua from GreatBritain
Category: Heavyweight
Wins: 19
Losses: 0
SPRINTER: Asaffa Powell from Jamaica
PERSONAL RECORDS:
100m: 9.72s
200m: 19.9s
SPRINTER: Tyson Gay from USA
NO PERSONAL RECORDS SET
SPRINTER: Usain Bolt from Jamaica
PERSONAL RECORDS:
100m: 9.58s
200m: 19.19s
BOXER: Wladimir Klitschko from Ukraine
Category: Heavyweight
Wins: 64
Losses: 5
#####
```