

CS1331 Homework 7 - World Cup

Introduction

In this assignment, you'll practice - implementing interfaces, - using generics, - and implementing a simple data structure.

Problem Description

You're suffering from the classic college student grind. While looking at the bleak dining hall options, you develop a plan to make some money on the World Cup. The first step to your elaborate scheme is to implement a `TeamSet` so that you can track players and teams.

Eventually you hope to develop your code into a complex betting system that'll keep track of custom brackets, and help you escape the perils of academia.

Solution Description

The classes below have been provided for you. You are responsible for implementing most of `TeamSet` to help you keep track of all of the teams participating.

TeamSet.java

Implements the `Set` interface.

- Has the following instance variables:
 - `private T[] backingArray`: the array in which all objects in this set are stored.
 - `private int size`: represents the number of objects currently stored in this set. This variable must be updated when objects are added to or removed from this set.
- Has one no-args constructor that sets the `backingArray` to be of length 10.

In addition to the provided Javadocs, be sure to look at the [API](#) to see all of the details for each methods expected behavior, some methods are expected to throw exceptions under certain circumstances.

You must provide implementation for the following methods:

- `public boolean add(E e)`
- `public boolean addAll(Collection<? extends E> c)`
- `public void clear()`
- `public boolean remove(Object o)`
- `public boolean contains(Object o)`
- `public boolean containsAll(Collection<? extends E> c)`
- `public boolean equals(Object o)`
- `public Iterator<E> iterator()`
- `public Object[] toArray()`

- `public int size()`
- `public boolean isEmpty()` Notice that the Set interface has more methods, however you are not required to provide implementation for `public <T> T[] toArray(T[] a)`, `public boolean removeAll(Collection<?> c)`, `public int hashCode()`, and `public boolean retainAll(Collection<?> c)`.
- You may not add any additional import statements, public methods, or instance variables to TeamSet.
- You may not use `System.arraycopy`.
- You must complete any missing/incomplete Javadocs for this class.

Player.java

This file represents a player, which you will be using to fill your TeamSet.

WorldCup.java

This tester has been provided for you. Feel free to supplement it with additional checks.

- Running our tester does not guarantee you a 100. We encourage you to write your own test cases to fully test your homework files.

Javadocs

You will need to write Javadoc comments and watch for checkstyle errors with your submission.

- Every class should have a class level Javadoc that includes `@author <GT Username>` and `@version <version number>`.
- Every public method should have a Javadoc explaining what the method does and includes any of the following tags if applicable:
 - `@param <parameter name> <brief description of parameter>`
 - `@return <brief description of what is returned>`

See the CS 1331 Style Guide on Canvas for details.

Checkstyle

For each of your homework assignments we will run checkstyle and deduct one point for every checkstyle error, with the points deducted being capped.

For this homework the **checkstyle cap is 100**, meaning you can lose up to 100 points on this assignment due to style errors. This limit will increase with each homework.

- If you encounter trouble running checkstyle, check Piazza for a solution and/or ask a TA as soon as you can!
- You can run checkstyle on your code by using the jar file found on Canvas that includes xml configuration file specifying our checks. To check the style of your code run `java -jar`

```
checkstyle-6.2.2.jar *.java.
```

- To check your Javadocs run `java -jar checkstyle-6.2.2.jar -j *.java`.
- Note that the command for checking code and the command for checking Javadocs are different. You will have to run both commands to fully test for style errors.
- Javadoc errors are the same as checkstyle errors, as in each one is worth a single point and they are counted towards the checkstyle cap.
- **You will be responsible for running checkstyle on ALL of your code.**
- Depending on your editor, you might be able to change some settings to make it easier to write style-compliant code. See the Customization Tips on Canvas for more information.

Collaboration

When completing homeworks for CS1331 you may talk with other students about:

- What general strategies or algorithms you used to solve problems in the homework
- Parts of the homework specification you are unsure of and need more explanation
- Online resources that helped you find a solution
- Key course concepts and Java language features used in your solution

You may not discuss, show, or share by other means the specifics of your code, including screenshots, file sharing, or showing someone else the code on your computer, or use code shared by others.

Examples of approved/disapproved collaboration:

OKAY: "Hey, I'm really confused on how we are supposed to implement this part of the homework. What strategies/resources did you use to solve it?"

BY NO MEANS OKAY: "Hey... the homework is due in like 20 minutes... Can I see your code? I *promise* won't copy it directly!"

In addition to the above rules, note that it is not allowed to upload your code to any sort of public repository. This could be considered an Honor Code violation, even if it is after the homework is due.

Submission

- Submit your `TeamSet.java` file for this homework as an attachment to the `Homework 7` assignment on Canvas. You can submit as many times as you want, so feel free to submit as you make substantial progress on the homework. We only grade your **last** submission, meaning we will ignore any previous submissions.
- If you submit multiple times Canvas will append a number to your zip file (`TeamSet.java` becomes `TeamSet-1.java`). **Do not worry about this, we will fix the file name before compiling and running your code.**
- Non-compiling code will be given a score of 0. For this reason, we recommend submitting early

and then confirming that you submitted ALL of the necessary files by re-downloading your file(s) and compiling/running them.

- Files that contain inappropriate language and/or profanity will receive a 0.

Good Luck! \ (°□°) /