

CSCD 211

Lab 3

PROGRAM SPECIFICATIONS

In this assignment you will start building the basis for a football team. Your team will be contained in an ArrayList. The information for a team will come from multiple files specified by reading the contents of the directory.

CLASS SPECIFICATIONS

Position

Values, kicker, quarterback, runningback, tightend, widereceiver

toString returns K, QB, RB, TE, WR

Player

A **Player** class contains the following attributes as **private** instance fields:

- First Name – String
- Last Name – String
- Team Name – String
- Position Type – Position Enum
- Games Played - int

The class should have the following instance behaviors:

- An explicit value constructor (passed the above instance fields)
- A toString (returns a String formatted as follows:
Name: first last
Team: the team name
Position: the position
Games Played: the number
- A compareTo method that compares by **Games Played, Team Name, Last Name, First Name**
- Accessor (get) methods for each field *as you deem necessary*
- Modifier (set) methods for each field *as you deem necessary*
- Helper methods *as you deem necessary*

CSCD211Lab3Methods

- `readDirectoryName(kb)` – Prompts the user to enter a directory name. Ensures the name entered is a directory if not reprompts. Once a valid directory name is obtained a File object is returned containing the directory name.
- `readDirectoryContents(dirName)` – Using the passed in file object this methods returns an array of file objects representing the files within the directory.
- `readPlayers(inFileNames)` – reads in the information from the file creates and array list and inserts the player into the array list. When all files have been processed the array list is returned. The list will be trimmed to size after the operation is complete.
- `menu(kb)` – valid menu choices are below. You must ensure the number is within range.
 1. Print all Players to the screen
 2. Print all Players to the User Specified file
 3. Sort the Players by "Natural Order"
 4. Sort the Players by Team and print to the screen
 5. Sort the Players by Position and print to the screen
 6. Remove a player
 7. Print a entire team and only that team to a user specified file
 8. Quit
- `printList(myPlayers, System.out)` – Prints the list one player at a time with a carriage return separating each player.
- `readFileName(kb)` – reads and returns a string representing the name of the output file
- `printList(myPlayers, fout)` - Prints the list one player at a time with a carriage return separating each player.
- `removePlayer(myPlayers, kb)` – prompts the user for player information. Searches the list for that player and removes the player from the array list. The list will be trimmed to size after the operation is complete. If no player can be found the appropriate message will be displayed and the list will remain unchanged.
- `printTeamPlayers(myPlayers, kb)` – prompts the user for the 3 letter team code and prints all the players from that team - one player at a time with a carriage return separating each player.

INPUT FILE SPECIFICATIONS

The position type will be derived from the csv file name. The input file will have at least one entry. Each player type is specified as a CSV file. Sample input files are provided.

OTHER SPECIFICATIONS

- You must validate all ranges
- Your ArrayList will always be of the correct size. (trimToSize)
- Since we are using ArrayLists we use Collections instead of Arrays. The second argument is still a class that implements a Comparator.
 - **`Collections.sort(myPlayers, new TeamSort());`**
 - **`Collections.sort(myPlayers, new PositionSort());`**

TO TURN IN

There are multiple turn ins for this lab.

1. Your stubbed out methods ensuring all code compiles will be submitted via git commit to your repository on GitHub. This is required within by 11 PM on (See Date Specified in Canvas).
2. A zip file, in Canvas, by the required due date containing:
 - all java files
 - all input file(s) used to test your program
 - all output file(s)
 - a test run named cscd211lab3out.txt – testing all aspects
 - We should be able to download the zip and compile your code, and then run your code.
 - Name your zip, your last name first letter of your first name lab3.zip (Example: steinerslab3.zip)