Group 7

Youth Hockey Cup

Object Oriented Programming Assignment on Python

Submitted by:

Abu Abdullah Dhrubo
Nahom Gebremeskel

Date: 24th March 2022

Subject: Python and R Programming

Table of Contents

Introduction

This project involves the development and implementation of an object-oriented program which is used to track all the teams that a user of the program establishes and manages for the Youth Hockey Cup. All the information is kept in a file named HockeyTeams.txt .

In this procedure, the user will be able to do a variety of tasks, as seen below.

- create new teams with attributes
- read a specific team using unique id to show its information
- update fields for a selected team
    - update one or all the fields based on team id
    - cancel their registration using update teams
- team details:
    - show the number of teams currently available
    - show the number of teams cancelled the registration
    - show the percentage of teams paid the registration fees
- delete a team
- list teams, list all registered girls' teams or boys' teams
- list all teams registered regardless of girls or boys
- save team details in a text file
- reading the text file to memory from HockeyTeams
- exit from the program

For the user to interact with the program and conduct essential actions, a menu control was established.

Description

We built Classes to make it more user-friendly by allowing us to code using Object Oriented Programming. Classes created are,

1. Main
2. Team
3. UserInterface

Let's go through what each of the ability classes is made up of, how they're utilized, and how they're connected.

Class Main

- The Main Class is the program that allows the user to display the required menu and interact with the UserInterface to complete the required activities.
- This is the section of the program that must be run; it imports essential modules and performs the tasks which user has asked. When we call the Main class, several functionalities are called from the Team and UserInterface class to execute the action that the User wanted.

Class Team

- The Team Class defines the structure for creating and storing team attributes, as well as the information needed for an object to be classified as a team.
- Aside from the required private attributes,
    1. We have set a fee that for registration as, “fee”.
    2. When a new team is created, a counter (Team.cnt) variable is used to generate a unique id.
    3. An id_list contains all the ids in a list that can be used to execute various actions.
    4. A team_list contains all the teams in a list that can be used to execute various actions.
- Accessor and mutator methods are provided for the attributes so that values assigned to the attributes can be accessed and updated from outside the class as needed.

- Additional methods were also developed to retrieve the complete information of the teams from outside the class.

Class UserInterface

- The UserInterface (UI) Class allows the program to take control of the user-selected input choice from the menu and call the appropriate function to accomplish the action.
- For each action to be performed, such as creating, updating, and deleting a team, separate methods were built.
- For accessing the teams that were created using the program can be stored and loaded, if needed.
- Also, the user can choose to exit if the actions are complete.

Additional features incorporated

- When the user, creates a new team it automatically saves the information in a text file also after updating the existing teams its saved in the text file automatically.
- When a team has been created the team id is unique and not repeated. Which is when the user loads the data from the text file, new entries will continue to be unique.
- Checks for the existence of the teams, if options other than creation, exit and load are selected it prompts that, “there are currently no teams registered to take such action”.
- Validated user input when team id is required as input for performing an action.

User Guide

Execute the main function (class_main()) to run with the program. Menu options will be presented to the user with a different choice from 1-10. The user decides which action to be executed.

• Option - 1 (Create or add a new Team): -
Program allows users to input: - the name of the team, team type (gender), and fee payment status. The program will automatically take into consideration the current date as the registration date after execution.

• Option - 2 (Show Individual Teams.): -
Displays list of available team id from the memory and lets the user choose which team they want to view by choosing team id.

• Option - 3 (Read Teams based on Type.): -
Takes input from user and print the list of all teams for boys or girls and show all information.

• Option - 4 (Read All Teams): -
Displays all information about the teams created or updated.

• Option - 5 (Update existing Team): -
Allows the user to select the team's id for which the fields are to be updated from the list of available teams. Then ask the user whether only one or all the fields to be updated. Based on the user's input the program will update team information and the text file will also be updated.

• Option - 6 (Delete existing Team): -
Allows the user to select the team's id which is to be deleted from the memory and the text file.

• Option - 7 (Show existing Teams details): -
Displays list of existing team's details. After execution, the number of teams registered, the number of teams canceled (if any), and the percentage of teams who paid fees will be presented respectively.

• Option - 8 (Save Teams in a Text file): -
Executing this option will store all the available teams' information in a text file in a readable format.

• Option - 9 (Load Teams from Text file): -
Import teams' information from previously saved text file and loads it into the memory for the users to execute different actions.

• Option - 10 (Exit): -
Exit option allows the user to exit the program with or without saving the data from the memory.

All options are fully functional and easily presented to the user. Options 2-8 check if any teams are present in the memory and proceed further with required actions if teams' information are present only.