Logo, Dalarna University

A logo of Dalarna University

**GIK29B: PYTHON AND R PROGRAMMING**

**HANDBALL TEAMS REGISRATION AND ADMISTRATION PROJECT**

**JAPHET ALFRED NTWALLE**

**2022-10-31**

Table of Contents

[1 Introduction 3](#_Toc118065684)

[2 Description 3](#_Toc118065685)

[2.1 Team class 3](#_Toc118065686)

[**2.1.1 Data attributes** 4](#_Toc118065687)

[**2.1.2 Setters, getters, string method** 4](#_Toc118065688)

[2.2 Interface class 4](#_Toc118065689)

[2.3 Main Program 6](#_Toc118065690)

[3 Conclusion 6](#_Toc118065691)

[4 User Guide 6](#_Toc118065692)

[4.1 How to run the program 6](#_Toc118065693)

[4.2 Create a team 7](#_Toc118065694)

[4.3 Show team using ID 7](#_Toc118065695)

[4.4 Show team by type 7](#_Toc118065696)

[4.5 Show all teams 7](#_Toc118065697)

[4.6 Update team 8](#_Toc118065698)

[4.7 Delete team 8](#_Toc118065699)

[4.8 Save data 8](#_Toc118065700)

[4.9 Read data 8](#_Toc118065701)

[4.10 Total number of teams 8](#_Toc118065702)

[4.11 Cancel participation 8](#_Toc118065703)

[4.12 Quit the program 8](#_Toc118065704)

# **1** **Introduction**

This report describes an object-oriented program implemented in the Python programming language. The program allows the user to register and administer handball teams for a handball event. It allows the user to create teams, and display team information based on team unique identifiers, names or types. It also allows the user to update different fields of a particular team, delete a team from the list, save all the team information as a string to a text file, restore all the information into the program as objects as well as display a summary about the teams.

The program contains 3 python script files: ***team.py, menu.py*** and ***interface.py.*** The team.py script contains one class named Team which is used to create teams which are instances of the class Team. The interface.py file defines one class named Interface with methods to create, read, update, delete, save and restore teams and display a summary of team registered. The menu.py file contains the main program which allows the user to run the program.

Since the program is to be executed on the command line interface, it assumes the user is familiar with basic Linux commands such as cd to navigate to the folder containing the files. The user will run the program using the command python menu.py and will be presented with a menu to perform all the required operations.

# **2** **Description**

In this section, a description of each class used in the program is presented. This includes the purpose of each class and how it interacts with other classes in the program.

## **2.1 Team class**

The Team class is contained in the team.py script file. The purpose of this class is to allow the user to create a handball team object containing a team id, date of creation, name, fee status, fee amount and team participation status. The Team class is imported into the Interface.py script to be used for creating, updating, reading, and deleting teams.

Text, website

a python class definition

Figure 1: Definition of the Team class

The class Team contains a class variable which is used to increment the id attribute for each object created. This is done to make sure each object created has a unique identification.

### **2.1.1 Data attributes**

The Team class is initialized by defining the *\_\_init\_\_* method and passing the data attributes as parameters. The id and date attributes are set automatically, and they are read-only as the user cannot change them. During the creation of the object the user only has access to the team’s name, type and fee status. The fee amount is set automatically depending on the fee status, if a team has paid the fee, the fee amount will be set to 99$ otherwise it will be set to 0. The cancel attribute represents the participation status of the team. All the teams created will have a default value of None with the assumption that all teams will be participating in the event. If a team cancels the attribute will be updated to a date representing the date a team cancelled their participation in the event



Figure 2: Initialization of a class object

### **2.1.2 Setters, getters, string method**

To restrict user access to the data attributes and validate the user input, the *@property* decorator is used. A string method is included to allow displaying of team information in a human-readable format. The id and date attributes do not have setter methods since they are not meant to be manipulated by the user. The rest of the. Attributes have getters to return the attribute with corresponding setters to validate the data

Graphical user interface, application

Description automatically generated

Figure 3: Example of a setter and getter using the @property decorator

## **2.2 Interface class**

The Interface class is defined in the interface.py script file and it contains methods to perform all the operations. The class define the methods that will be used to create a team, read and update team information, delete a team from the list, save the team information as strings to a text file, restore the team information as objects using the saved text file and display a summary of all the teams registered. Also, the Interface class has a class attribute named teams which is a list variable responsible for storing all the teams created.

The Interface class imports the Team class from the team.py file and use it to perform the following operations:

o **Create team objects** (instances of the Team class): Allow the user to create a team by passing in the team name, type and fee status, the method then create a team and append it to the list of teams

o **Display team information:** Allow the user to display team information using the team id. The user enters a team id and the methods display the team id, name, type, fee status, fee amount and participation status. The user can also choose to display teams by type or display all teams contained in the list

o Update team information: By using the team id, the user can update the team name, type, fee status or participation status.

o **Cancel participation:** If a team withdraw from the event, the team information is updated to include the date the team cancelled their participation. By default, all teams created have a default value of None meaning that they will be participating in the event

o **Save data:** After creating teams, a user can decide to save the teams' information in a text file to have access to them even after the program session has ended. This will allow the user to restore the information the next time they run the program and be able to add, display, update or delete teams. All the teams are stored in a text file named my\_teams.txt

o **Read data:** if the user has previously created teams and saved them into a text file, this method allows them to restore the teams into the program. By using the text file, the teams are recreated back into objects and then stored in a list. The user can then continue to perform other operations as usual.

o **Count teams:** displays a summary of all the teams registered for the event, it shows the total number of teams registered for the event, the percentage of teams which have paid the participation fee as well as the total amount of fees paid

o **Display the menu:** For the user to be able to run the program, two methods are defined to display the program menu, one to display the main menu and the second to display a sub-menu when updating a team information

o **Delete team: This** will remove a team from the list of teams. It uses the team id to identify which team to remove from the list

## **2.3 Main Program**

The menu.py script file is the main program/executing program. It imports the Interface class and calls its methods accordingly. It also imports the sys module that is used to exit from the program. The program has global constants for the main menu and the update sub-menu. Once the user runs the program, the main menu will be displayed, and the user will be prompted to enter their choice according to the menu. The user executes the program by using the command python menu.py on the command and exits the program by entering 0 on the command line prompt.

# **3 Conclusion**

This report described a python program for registering and administering handball teams. The program is executed from the command line interface using the menu.py script file. After navigating to the folder containing all the program files, the command python menu.py is run from the command line to start the program. The user then follows the instructions from the program menu to perform all the operations as well as to exit the program.

# **4 User Guide**

This program allows the user to register and administer handball teams. The program contains three python script files named team.py, interface.py and menu.py. To be able to run the program make sure that all the files are in the same folder/working directory.

## **4.1 How to run the program**

* Open the terminal/command line
* Navigate to the folder where the scripts are stored
* Run the command python menu.py from the terminal: After running the command a menu will be displayed as well as a prompt to enter a choice

Text

Description automatically generated

* Your choice should be an integer from the main menu

## **4.2 Create a team**

On the command line prompt enter 1 followed by the enter key. Fill in the name of the team to be created, the type of team ( B for boys and G for girls) and fee status to show if a team have paid the participating fee ( yes / no)



## **4.3 Show team using ID**

Display team information using the team id. Enter 2 followed by the enter key on your keyboard. Enter the id of the team to display and hit enter. The team info will be displayed above the main menu

A picture containing text

Description automatically generated

## **4.4 Show team by type**

Display team information based on type. Enter 3 followed by the enter key. Fill in the type of team to show (b/boys, g/girls). This will display all teams of the specified type.

## **4.5 Show all teams**

Enter 4 followed by the enter key. This will display all the teams created so far. Since the teams are displayed in rows, make sure your command line display window is expanded to see all the teams.

## **4.6 Update team**

Enter 5 followed by Enter key. A sub-menu to update teams will be displayed followed by a prompt on what field to update. Enter the corresponding number from the sub-menu and fill in the next prompts

Text

Description automatically generated with medium confidence

## **4.7 Delete team**

Enter 6 followed by the enter key. A prompt will be displayed, enter the id of the team you will like to display, if there’s no team with such id, the program will return to the main menu.

## **4.8 Save data**

Enter 7 followed by the enter key. This will save all the teams created into a text file named my\_teams.txt.

## **4.9 Read data**

Enter 8 followed by the enter key. This will restore all the teams previously stored in the text file. If the team were not saved, the program will return to the main menu.

## **4.10 Total number of teams**

Enter 9 followed by the enter key: This will display a summary of the team registered. The total number of teams registered the percentage of teams which have paid the participation fee and the total amount of fees paid.

## **4.11 Cancel participation**

Enter 10 followed by the enter key. This will display a prompt to enter a date in the format YYYY-MM-DD, this is the date when the team cancelled their participation in the event.

## **4.12 Quit the program**

To exit the program, enter 0 followed by the enter key