THE SPORTS EVENT PROGRAM

2018-11-08

Course Code: IK1052

Course Name: Introduction to Object-oriented programming

Course Manager: Roger Nyberg

Semester/period: HT 2018

Author: Raja Umer Saghir

THE SPORTS EVENT PROGRAM

Table of content

1. Introduction
2. Description of solution
3. Classes and collaboration
4. User guide
5. Comments.

1. Introduction

I am supposed to make an object-oriented program for the sports event which will keep order of the teams participating in the and sports event. This program will keep track of all the teams participating in the event.

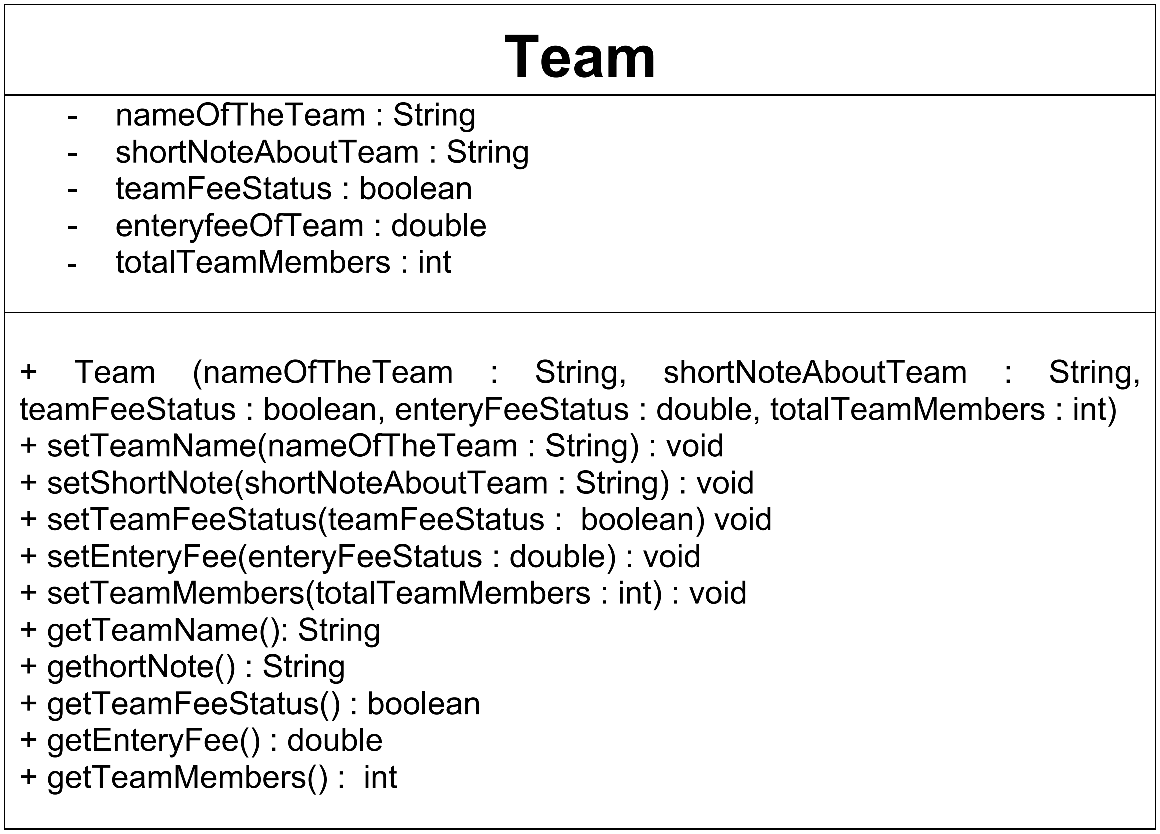
This program will have following characteristics.

- The program will provide the user interface to add the data of the teams will by presenting menu option which includes.

    - Name of the team
    - A short note about the team
    - If the team has paid the entry fee
    - Enter fee amount
    - The number of team members.

- Program will check the validity of the entered data.
- It will allow to remove any team after adding.
- Program will check the make possible to make changes in some data with the help of displayed menu and choices.
- Program will prevent changing some data like name because it will remain the same.
- It will provide the display to see the information stored regarding the team.
- It will allow user to repeat the steps on selection of choice.
- Allow user to quit the program at different stages with help of displayed menu and choices.

2. Description of the solution



Team

- nameOfTheTeam : String
- shortNoteAboutTeam : String
- teamFeeStatus : boolean
- enteryfeeOfTeam : double
- totalTeamMembers : int

+ Team (nameOfTheTeam : String, shortNoteAboutTeam : String, teamFeeStatus : boolean, enteryFeeStatus : double, totalTeamMembers : int)
+ setTeamName(nameOfTheTeam : String) : void
+ setShortNote(shortNoteAboutTeam : String) : void
+ setTeamFeeStatus(teamFeeStatus : boolean) void
+ setEnteryFee(enteryFeeStatus : double) : void
+ setTeamMembers(totalTeamMembers : int) : void
+ getTeamName(): String
+ gethortNote() : String
+ getTeamFeeStatus() : boolean
+ getEnteryFee() : double
+ getTeamMembers() : int

Team Class

Fields

- The field with name nameOfTheTeam is String and will be used to store the name of the teams.
- This field with name shortNoteAboutTeam is String will be used to store the short note of teams.
- This field with name teamFeeStatus is boolean will be used to store the status of the team's fee.
- This field with name enteryfeeOfTeam is double will be used to store the fee paid by teams.
- This field with name totalTeamMembers is double will be used to store the total team members in the team.

Constructer

- This will initialize the values of the all field of Team class when object is created.

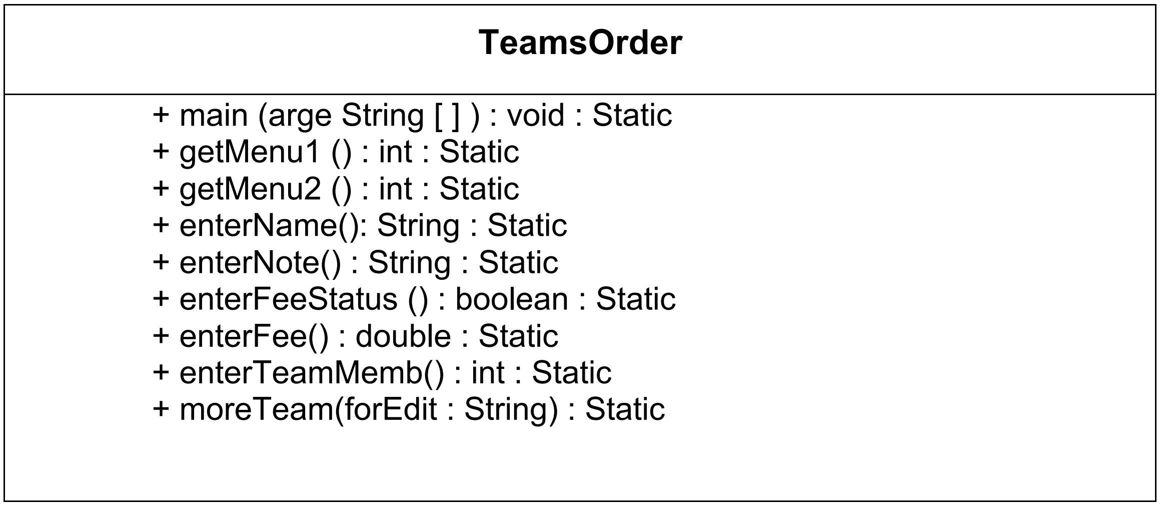
Mutators

- The mutator with name setShortNote is of type void and accept the String arguments and will set note for the team when user want to change.
- The mutator with name setTeamFeeStatus is of type void and accept a boolean arguments and will set the value of the team's fee status when user want to change.
- The mutator with name setEnteryFee is of type void and accept double arguments and will set team fee amount when user want to change it.
- The mutator with name setTeamMembers is of type void and accept int arguments and will set the value of the team members.

Accessors

- The accessor with name getTeamName is of return type String and accept no arguments and will get name of the team when user want to access.
- The accessor with name getShortNote is of return type String and accept no arguments and will get short note of the team when user want to access.
- The accessor with name getTeamFeeStatus is of return type boolean and accept no arguments and will get fee status of the team when user want to access
- The accessor with name getEnteryFee is of return type double and accept no arguments and will get fee amount of the team when user want to access.
- The accessor with name getTeamMembers is of return type int and accept no arguments and will get members of the team when user want to access.

TeamsOrder Class



TeamsOrder

+ main (arge String [ ]) : void : Static
+ getMenu1 () : int : Static
+ getMenu2 () : int : Static
+ enterName(): String : Static
+ enterNote() : String : Static
+ enterFeeStatus () : boolean : Static
+ enterFee() : double : Static
+ enterTeamMemb() : int : Static
+ moreTeam(forEdit : String) : Static

Methods

Main Static Method

- The Main method will be user interface with all menu.
- The Main method will be used to create objects Team class.
- The Main method will use Arraylist to store all objects.
- The Main method will control all the operations of program.
- The main method will set the values of the team objects in team class fields with the help of mutators.
- The main method will get the values of the team objects from Team class fields with the help of accessors.
-

Other Static Methods

- The method with name getMenu1 is a static type and return type int which accept no arguments and will used to display first menu for user with two choices either enter the data or quit.
- The method with name getMenu2 is a static type and return type int which accept no arguments and will used to display second menu for user with four choices for display information, want to remove team, want to update and quit.
- The method with name enterName is a static type and return type String which accept no arguments and used to change the name of the teams.
- The method with name enterNote is a static type and return type String which accept no arguments and used to change the note of the teams.
- The method with name enterFeeStatus is a static type and return type boolean which accept no arguments and used to change the status of the teams.
- The method with name enterFee is a static type and return type double which accept boolean arguments returned from enterFeeStatus and used to change fee of the team.
- The method with name enterTeamMemb is a static type and return type int which accept no arguments and used to change total team members.
- The method with name enterTeamMemb is a static type and return type char which accept String arguments and used for add more teams which depend on user choice.

4. User Guide

Step 1

1. A welcome note will be displayed.
2. A menu will be displayed with two choices
3. Enter 1 for entering the data and 2 to quit the program.

Step 2

1. A menu will be displayed for entering the data.
2. First user needs to enter the name of the team.
3. Then user need to enter the note for the team.
4. Then user need to choose between if team has paid fee.
5. If user will choose not paid then will not ask for amount of fee and 0.0 will be added. (Y or y for yes and n or N for no)
6. If user will choose yes for the fees, then user will be prompted to enter the fee amount.
7. Then user will enter the team's member in the team.
8. Then user will be prompted to choose if he/she want to add more teams. (Y or y for yes and n or N for no)
9. If user will choose yes, then step 2 will be repeated.
10. If user will choose no, then will enter in next menu with 4 choices.

Step 3

1. If user choose 1 then program will display all the team with names.
2. Then user will be prompted to select the numbers according to team for which he/she want to display.
3. Then information will be displayed according to user choice.
4. If user choose 2 then program will be prompted to choose the team for which he/she what to remove information.
5. If user choose 2 then program will be prompted to choose the team for which he/she what to remove information.
6. Then user will be prompted to choose which team he or she want to remove.
7. After choosing user that team will be removed, and user will be prompted to select if he/she want to see the menu again.
8. If user choose 1 then menu will be displayed again.
9. If user will choose 3 then teams name list will be displayed, and user will be asked to which team he or she want to remove data.
10. If user will choose the team number, another menu will be displayed asking user that what data he want to update.
11. Choosing 1 user will be prompted to enter the new note to change the note in new menu.
12. Choosing 2 user will be prompted to enter the new fee to change the fee in new menu and status will be updated as well.

13. Choosing 3 user will be prompted to enter the new total team members to change total members in new menu.
14. After adding that user will be prompted if he wants to make more changes.
15. If choosing yes, then user will be displayed with the new menu.
16. If choosing no the user will be prompted to choose if he want to change see the 4 choices menu.
17. If user will choose not, (2) then user will be prompted if he want to start the program from the start.
18. If user choose not (1) then program will quit.
19. After displaying the information user will be prompted to enter the choice for continue working with the program or quit.

5. Comments

1. All the work in commented in the program.