

User Class

Variables:

- **userName (String):** The name of the user.
- **userType (String):** The type of user, which could represent different roles or permissions in a system.
- **userID (int):** A unique identifier for the user.
- **activityList (ArrayList):** A list of activities associated with the user.

Methods:

1. **Default Constructor (no parameters)** Initializes a new User object with default values: null for userName and userType, 0 for userID, and an empty ArrayList for activityList.
2. **Main Constructor (with parameters)** Takes userName, userID, and userType as parameters and initializes a User object with these values, provided that the name is valid. Also initializes the activityList.
3. **Get Method for userName** Returns the user's name as a string.
4. **Get Method for userType** Returns the user's type as a string.
5. **Get Method for userID** Returns the user's ID as an integer.
6. **Get Method for activityList** Returns the user's list of activities as an ArrayList.
7. **Set Method for userName** Takes a string as a parameter and sets the userName if it is currently valid to do so.
8. **Set Method for userType** Takes a string as a parameter and sets the userType if it is currently valid to do so.
9. **Set Method for userID** Takes an integer as a parameter and sets the userID.
10. **Set Method for activityList** Sets the activityList to a new ArrayList, if the current list is null.
11. **addActivity (method)** Adds an Activity object to the activityList.
12. **removeActivity (method)** Removes an Activity object from the activityList.
13. **isNameValid() (function)** Checks whether the userName has not been set and thus can be set.
14. **isTypeValid() (function)** Checks whether the userType has not been set and thus can be set.
15. **isIDValid() (function)** Checks whether the userID is in a specific valid state (in this case, the method checks for a hardcoded value which seems incorrect).
16. **toString() (function)** Returns a string representation of the User object, including the user's information and a list of their activities. If there are no activities, it indicates "None".

Activity Class

Variables:

- **activityName (String):** The name of the activity.
- **activityTime (Date):** The scheduled time for the activity.
- **activityVenue (String):** The venue where the activity will take place.
- **maxPlayerNumber (int):** The maximum number of players that can participate in the activity.
- **memberList (ArrayList):** A list of users who are members of the activity.

Methods:

1. **Default Constructor (no parameters)** Initializes a new Activity object with default values: "TBD" for activityName and activityVenue, the current date for activityTime, and 0 for maxPlayerNumber.
2. **Main Constructor (with parameters)** Takes activityName, activityTime, activityVenue, and maxPlayerNumber as parameters to initialize an Activity object. Also initializes the memberList with an empty ArrayList.
3. **Get Method for activityTime** Returns the activity's scheduled time as a Date.
4. **Get Method for activityName** Returns the activity's name as a string.
5. **Get Method for activityVenue** Returns the activity's venue as a string.
6. **Get Method for maxPlayerNumber** Returns the maximum number of players for the activity as an integer.
7. **Get Method for memberList** Returns the list of members (users) as an ArrayList.
8. **Set Method for activityTime** Takes a Date as a parameter and sets the activityTime if it is currently valid to do so.
9. **Set Method for activityName** Takes a string as a parameter and sets the activityName.
10. **Set Method for activityVenue** Takes a string as a parameter and sets the activityVenue if it is currently valid to do so.
11. **Set Method for maxPlayerNumber** Takes an integer as a parameter and sets the maxPlayerNumber if it is currently valid to do so.
12. **addMember (method)** Adds a User object to the memberList if the number of members is valid.
13. **removeMember (method)** Removes a User object from the memberList.
14. **isVenueValid() (function)** Checks whether the activityVenue has not been set and thus can be set.
15. **isMaxNumValid() (function)** Checks whether the maxPlayerNumber has not been set and thus can be set.
16. **isMemberNumValid() (function)** Checks whether the memberList is empty and thus new members can be added.
17. **isTimeValid() (function)** Checks whether the activityTime has not been set and thus can be set.
18. **toString() (function)** Returns a string representation of the Activity object, including the activity's information and the number of booked members.

App Class (main)

Variables

- **activityList**: A static list that holds all the activities.
- **users**: A static list that contains all the users.
- **uid**: A static integer that is used as the unique identifier for new users.

Main Method

The main method initializes the application, displays the main menu, and handles user input to navigate through different functionalities.

Main Functionalities The application provides several key features which are accessible from the main menu:

1. **Display Main Menu (displayMainMenu)** Displays the main menu options to the user, which includes the following: [D]: Display activities. [U]: Display all users. [S]: Search for an activity by time. [T]: Create a new team activity with settings. [P]: Personal user joins an activity. [C]: Personal user checks own activities. [R]: Personal user removes own activities. [O]: Organizer adds user to activity. [A]: Add a new user. [X]: Exit the application.
2. **Display Activities (displayActivities)** Lists all the activities currently in activityList with their details.
3. **Display All Users (displayAllUsers)** Shows all the users currently in the users list with their details.
4. **Search Activity (SearchActivity)** Allows the user to search for activities by specifying an activity time.
5. **Create Activity (createActivity)** Enables the creation of a new activity with details such as name, time, venue, type, and max number of members.
6. **Add User (addUser)** Allows adding a new user with a unique ID.
7. **Join Activity (joinActivity)** Facilitates a user to join an existing activity from the activityList.
8. **Add User to Activity (addUserToActivity)** Enables an organizer to add a user to an activity, increasing the number of participants.
9. **Check Personal Activity (checkPersonalActivity)** A user can check their own list of activities they have joined or created.
10. **Remove Personal Activity (removePersonalActivity)** A user can remove an activity from their personal list.