MIDTERM ASSIGNMENT

Subject : Classes, objects, encapsulation, inheritance, and polymorphism.
Instructor: Dr. Selim Yılmaz (selimyilmaz@mu.edu.tr) & Dr. Özgür Kılıç (ozgurkilic@mu.edu.tr)
Out Date: 05/03/2021 23:59:59
Due Date: 05/17/2021 23:59:59

DECLARATION OF HONOR CODE[1]

Student ID 170709012
Name Kıymet Deren
Surname TOY

In the course of Introduction to Object Oriented Programming (CENG 1004), I take academic integrity very seriously and ask you to do as well. That's why, this page is dedicated to some clear statements that defines the policies of this assignment, and hence, will be in force. Before reading this assignment booklet, please first read the following rules to avoid any possible violation on academic integrity.

- This assignment must be done individually unless stated otherwise.
- You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, you cannot copy code (in whole or in part) of someone else, cannot share your code (in whole or in part) with someone else either.
- The previous rule also holds for the material found on the web as everything on the web has been written by someone else.
- You must not look at solution sets or program code from other years.
- You cannot share or leave your code (in whole or in part) in publicly accessible areas.
- You have to be prepared to explain the idea behind the solution of this assignment you submit.
- Finally, you must make a copy of your solution of this assignment and keep it until the end of this semester.

I have carefully read every of the statements regarding this assignment and also the related part of the official disciplinary regulations of Muğla Sıtkı Koçman University and the Council of Higher Education. By signing this document, I hereby declare that I shall abide by the rules of this assignment to prevent any violation on academic integrity.

Signature .....

[1]This page should be filled and signed by your handwriting. Make it a cover page of your report.

MIDTERM REPORT

CENG 1004, INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

Kıymet Deren Toy
kiymetderentoy@posta.mu.edu.tr

May 14, 2021

1 Introduction

In this assignment, a social media platform has been created using the basics of object oriented programming with Java programming language. This platform basically includes 4 classes: User, Post, Location and MySocialBook. These classes have separate properties and methods. In summary, users created from the User class can have posts. Posts, on the other hand, must definitely have location. And the Post class is subclassified according to their properties.

2 UML Diagram

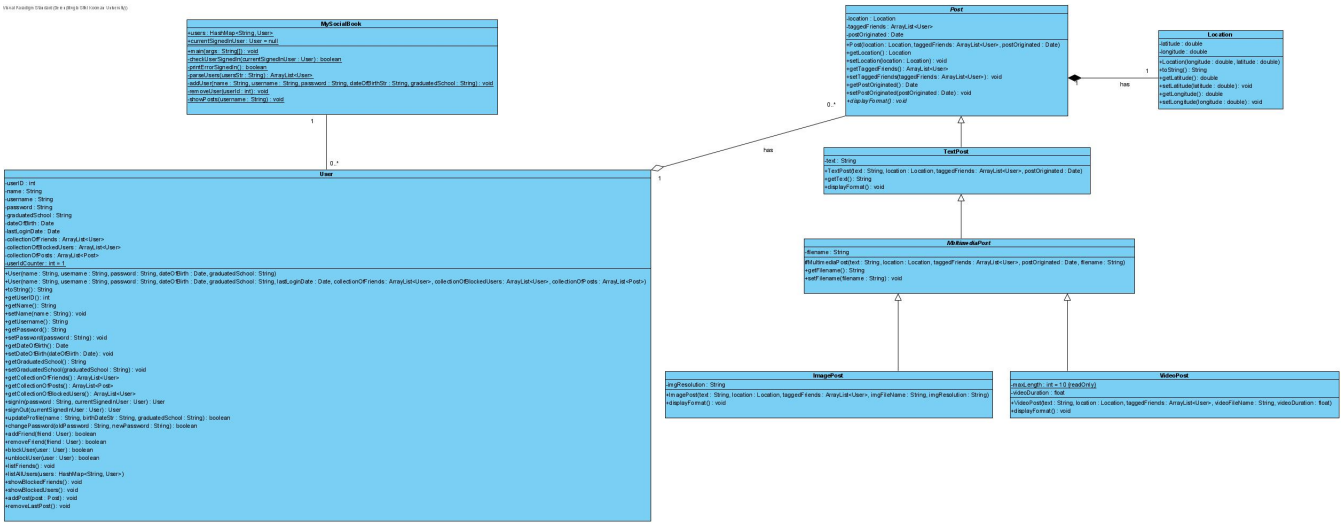

Figure 1: UML Diagram

3 Classes

3.1 User

The User class is located inside the User package. It keeps the properties and methods of the users in the system. All properties are private. There are two different constructors inside. It has 14 methods and the necessary getters and setters. Objects created from the User class are the people who will use the system.

3.2 Post

The Post class is included in the post package. It is an abstract class. It has 3 properties, a constructor with these and getter-setter methods of these properties. There is an abstract method named displayFormat. Objects created from the User class have Posts. Posts created from the Post class need location information created from the Location class.

1. **TextPost:** TextPost class inherits from Post class. For this reason, it has to implement the displayFormat method, which is abstract in the Post class. Unlike the properties in the post class, it includes a property that holds text.
    (a) **MultimediaPost:** This class is an abstract class and inherits from the TextPost class. In addition to the properties of the TextPost class, MultimediaPost class contains a property that holds the file name.
        i. **ImagePost:** ImagePost class inherits from MultimediaPost class. It was created for image posts. In addition to the MultimediaPost class, it includes a property that keeps the image resolution. And it overrides the displayFormat method in the TextPost class.
        ii. **VideoPost:** VideoPost class inherits from MultimediaPost class. It was created for video posts. In addition to the MultimediaPost class, it includes properties that keeps the duration of video and maximum length of video. And it overrides the displayFormat method in the TextPost class.

3.3 Location

The Location class is located under the location package. It includes two properties, longitude and latitude. It has a constructor, getter and setter methods built with these properties.

3.4 MySocialBook

The MySocialBook class, which is the tester class in the assignment, takes the users.txt and commands.txt files as arguments. And it allows testing the User class according to these files. It contains 6 static functions. And with the switch-case block it contains, it enables the methods in the User class to be called according to the commands read from the file.

4 Methods and Functions

4.1 User Class's Methods

1. **signIn:** This method works if the username entered exists in the HashMap where users are kept. Here, the password taken as a parameter is checked and if it matches with the password in the system, the user logs into the system.
2. **signOut:** This method allows the user to log out of the system by assigning the current user object to a null value.
3. **updateProfile:** This method sets the user's personal information such as name, date of birth, school information according to the entered values.
4. **changePassword:** This method allows the user to change their password after checking the old password entered.
5. **addFriend:** This method allows the user to add a new friend if the person they want to add as a friend is not his/her friend.
6. **removeFriend:** This method allows the user to remove anyone on the user's friend list from their friends list.
7. **blockUser:** This method allows blocking any user on the system by adding user to collection of blocked user list on the User class.
8. **unblockUser:** This method allows to unblock any user on the system by removing user from the collection of blocked user list on the User class.
9. **listFriends:** This method allows users in the user's friend list to be listed.
10. **listAllUsers:** This method ensures that all users on the system are listed.
11. **showBlockedFriends:** This method allows the user to view their blocked friends.
12. **showBlockedUsers:** This method allows the user to view blocked users.
13. **addPost:** This method allows the user to add all kind of posts to the system.
14. **removeLastPost:** This method allows the user to remove of their last post.

4.2 Post Class's Methods

1. **displayFormat:** This method determines the format in which Posts will be printed on the screen. Since this method is an abstract method, any new post type created from the Post class must implement this method.

4.3 MySocialBook Functions

1. **checkUserSignedIn:** This function checks whether users are logged in to use some method such as updateProfile, changePassword etc.
2. **printErrorSignedIn:** This function prints the error message if users tried to use User's method without logging in.

3. **parseUsers:** This function allows a colon-separated tagged string of users to be comma-separated and added to a list.
4. **addUser:** This function creates a user object based on the information entered and adds them to the users Hashmap.
5. **removeUser:** This function ensures that if there is a user registered in the system according to the entered id number, this user is deleted from the system.
6. **showPosts:** This function shows if the user has any posts, using the user's username.

4.4 Common Methods

1. **Getter:** This method provides access to private variables.
2. **Setter:** This method allows the values of private variables to be changed.
3. **toString:** This method is a method of the Object class, and it allows me to override it and print the objects in the form of the string I want.

5 Conclusion

This assignment allowed me to improve myself in object oriented programming. In this assignment, I experienced creating packages by paying attention to class hierarchy, building necessary constructors, creating class, method, abstract class, abstract method. It was a good experience for me to create the Post class that has subclasses and perform the operations related to this class.

Note: I added my UML diagram's picture to the zip file because I think that it is so tiny in report.

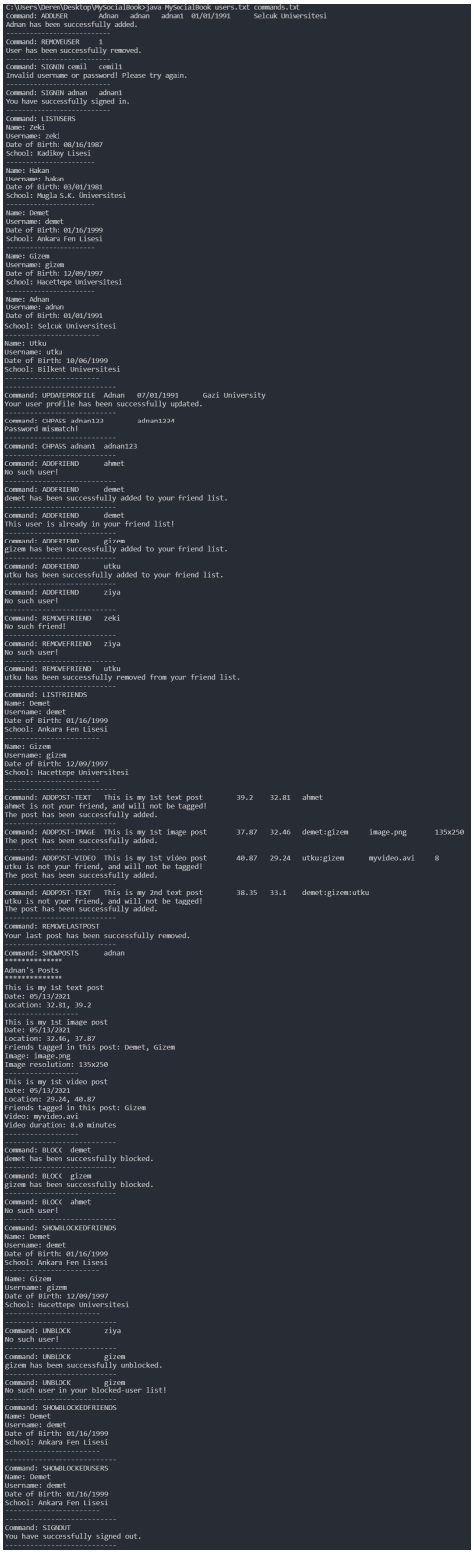

Figure 2: Output