**Object Oriented Development**

Module 9 : Inheritance - Classes

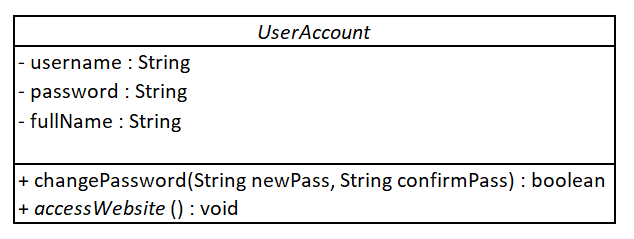
© FDM Group Ltd 2020. All Rights Reserved.

Any unauthorised reproduction or distribution in part  
or in whole will constitute an infringement of copyright.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 22 / 10 / 20 | Nick Lawton | First draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Part 1 – Creating an abstract class

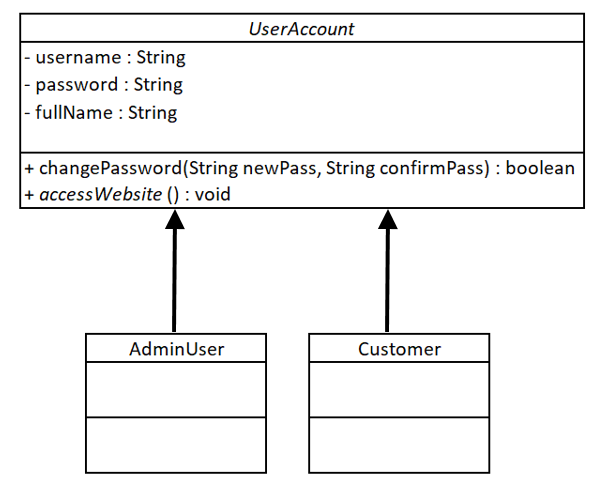
1. Create a package called com.fdmgroup.inheritanceExercises.
2. In the package, create the abstract class shown in the UML below (accessWebsite is an abstract method)



1. Create a constructor to initialise all 3 variables.
2. Create getters and setters for username and fullName. For password, just create a getter. The password will be changed via the changePassword() method.
3. Write the code in the changePassword() method. The method should check that the 2 String arguments match. If they do, it should change the password and return true. Otherwise it should return false.

## Part 2 – Creating concrete child classes

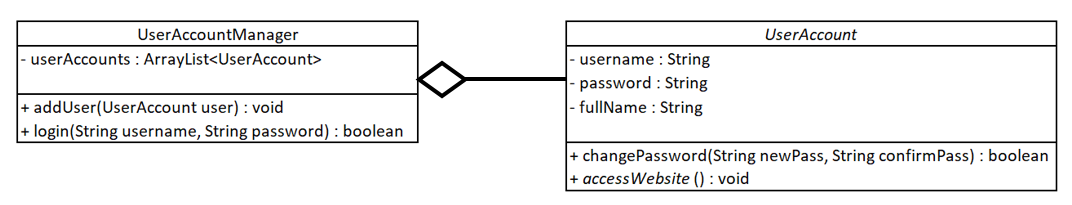
1. Create the concrete child classes shown in the UML below:



1. Write the code in the accessWebsite() method for both classes. You just need to put a simple message like ‘accessing website with admin rights’ or ‘accessing website as customer’.
2. Create a class called Runner with a main method.
3. In the main method, create 2 AdminUser objects and 2 Customer objects. Give them full names and usernames of your choice.
4. Call the changePassword() method to check that it works. Do this twice, once with arguments that match and once with arguments that don’t match.
5. Call the accessWebsite() method on an AdminUser object and on a Customer object. Check that it produces the right message for each.

## Part 3 – Creating a class which depends on the abstract class

1. Create the UserAccountManager class shown in the UML below. The userAccounts ArrayList should be instantiated within the class. Do not create getter and setter methods for it.



1. Write the code in the addUser method. The method should take the User argument and pass it to the userAccounts ArrayList’s add() method.
2. Write the code in the login() method. The method should loop through the userAccounts ArrayList until it finds a user account whose username and password match the username and password arguments. It should return true if a match is found and false otherwise.
3. In the main method of the Runner class, create a UserAccountManager object.
4. Call the UserAccountManager’s addUser() method 4 times – once for each of your 4 UserAccount objects.
5. Verify that the login() method works by calling it with a correct username and password as well as with invalid credentials.