**Object Oriented Development**

Module 11 : Polymorphism

© FDM Group Ltd 2020. All Rights Reserved.

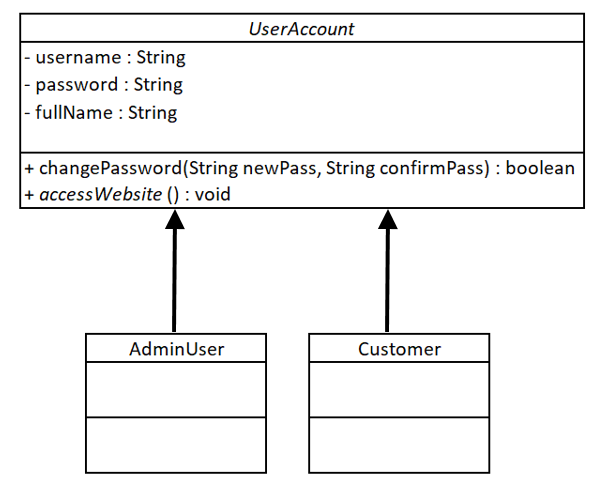
Any unauthorised reproduction or distribution in part  
or in whole will constitute an infringement of copyright.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 28 / 10 / 20 | Nick Lawton | First draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

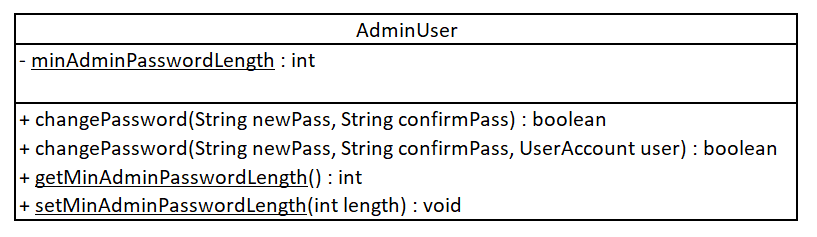
## Part 1 – Setting up

Create a package called com.fdmgroup.polymorphismExercises. Copy all the classes from your inheritanceExercise package (module 9).

Below is a reminder of the class hierarchy you created in the module 9 exercises.



In this exercise you’ll be overriding and overloading the changePassword() method of the AdminUser class, as shown in the UML below:



## Part 2 – Overriding the AdminUser changePassword() method

In this part, you’re going to write an overriden version of the changePassword() method which sets a minimum password length for AdminUser objects.

1. In the AdminUser class, create the **static** minAdminPasswordLength as shown in the UML above. Create **static** getters and setters for it.
2. In the main method of your Runner class call the setMinAdminPasswordLength() method directly from the AdminUser class, passing in the number 10.
3. In the AdminUser class write the overriden version of the changePassword() method. As well as checking that the 2 String arguments match, it should also check that their length is greater than or equal to the minAdminPasswordLength variable.
4. In the main method of your Runner class, call the changePassword() method from one of your AdminUser objects. Verify that the password is only changed when Strings of at least 10 characters are passed in.

## Part 3 – Overloading the AdminUser changePassword() method

In this part, you’re going to write an overloaded version of the changePassword() method which allows an AdminUser account to change the password of any UserAccount object.

1. In the AdminUser class, create the overloaded version of the changePassword() method. This time it should have a third argument: a UserAccount object.
2. Write the code in the overloaded method so that it calls the changePassword() method of the UserAccount object, passing in the 2 String arguments.
3. Verify that your code works by calling the overloaded changePassword() method on one of your AdminUser objects passing in 2 equal Strings and one of your Customer objects.

## Part 4 – Overriding the Customer changePassword() method

Follow the same logic you used in part 2 to create a minimum password length of 7 for the Customer class.