

PHP Classes

Create an abstract class User that will contain the following protected properties: firstName, lastName, username, password, role. The values of these properties need to be set with a constructor.

Make a method userPrint() that won't be inherited in the classes that will extend User class. This method returns a text in the following format: "first name: \$firstName, last name: \$lastName, username: \$username, role: \$role". You also need to define two more abstract methods: login() and logout();

Create a class Member that extends the User class. In addition, this class needs to contain the following properties: posts(protected), authenticated(private). The class constructor needs to set the value of the property posts to an empty array and the value of the property authenticated to false. You also need to call the constructor of the User class.

Create a login() method that will read from file users.txt file. The content of users.txt contains users in the following format: "username:password".

The login() method should set the authenticated property to true if the username/password combination is found in the users.txt file.

Create a method isLoggedIn() that will show the authenticated status.

Create a method getRole() that will return the member role.

Create a method logout() that will set the value of authenticated property to false.

Create a method identity() that will return the following text: "first name: \$firstName, last name: \$lastName, username: \$username, role: \$role, authenticated: \$authenticated".

Create an interface TopicInfo that will contain the following methods: printTopic() and printPosts()

Create a class Post that will contain the following private properties: createdTime, category, title, text. The class constructor needs to save the time of creation. For the other properties create getters and setters.

Create a class Topic that will implement the TopicInfo interface.

The class will contain the following private properties: title, author(object from Member class), category, createdTime and posts. The posts property will contain an array of associative arrays of objects from the Post and Member classes.

Eg: [

```
    ["post" => Post(...), "author" => Member(...)],  
    ["post" => Post(...), "author" => Member(...)],  
    ...  
]
```

In the constructor you need to check if the author - that is an incoming parameter in the constructor - is authenticated and his role is admin otherwise print error: "User: " . author->identity() . "is not able to create topic with title: \$title".

Create a method addPost that expects two input parameters: author and post. This method will check if the author is authenticated, has a role "MEMBER" and if the category of the post is the same with the category of the topic.

If everything is okay you need to save these input parameters in the posts array. Otherwise print the following error: "User: ". \$author->identity() . " is not able to add a post with title: " . \$title

Create a method printTopic() that will return a text in the following format: "Topic" . \$title.
"created by: ". author->identity() . " creation date: "\$createdTime

Create a method printPosts() that will return all posts in the following format: "Post: ".
\$post->getTitle() . " created by:". author->identity() . " date: ". \$post->createdTime