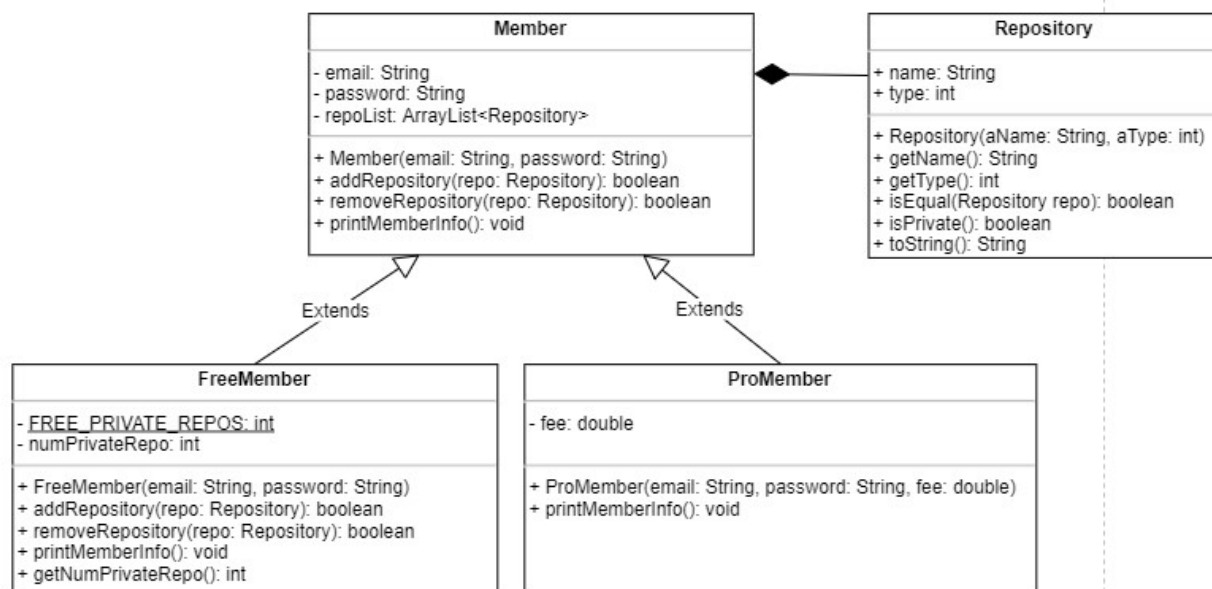


ITCS 209 Object Oriented Programming	Name:	Lab	Challenge Bonus	Peer Bonus
	ID:			
	Sec:			

### Lab06: Inheritance

GitHub is a web-based hosting service for version control using Git. It is mostly used for computer source code. In this lab, you will implement a program to manage GitHub's member accounts. Suppose that GitHub offer two types of member: Free and Pro member. Free members can store up to 3 *private* repositories and unlimited *public* repositories. (note: repository is a place to store the source code) Pro members can store unlimited *private* and *public* repositories. However, Pro members have to pay a monthly membership fee.

The `Repository` class and the `MemberTester` class are already provided. Please create three classes (`Member`, `FreeMember`, and `ProMember`) to practice your skills on inheritance concepts.



#### Member Class

`addRepository(Repository repo)`: If the repo is not null, this method adds that repo into the `repoList` and returns true. If repo is null, return false.

`removeRepository(Repository repo)`: If the repo is found in the `repoList`, removes it from the list, prints a successful message, and returns true. If the repo is not found in the `repoList`, returns false.

`printMemberInfo()`: Print information including email, password, and list of repositories (Please see the expected output for the exact format)

#### FreeMember class (This class extends Member class)

`FREE_PRIVATE_REPOS` is the maximum number of private repositories (the value = 3)

#### Overriding the following methods

`addRepository(Repository repo)`: If the repo is a private and the number of private repositories in the `repoList` is reaching the limit, prints a warning message and return false. Otherwise, adds the repo and returns true.

`removeRepository(Repository repo)`: If the repo is a private and found in the `repoList`, removes it from the list, decreases the number of private repositories, and returns true.

`printMemberInfo()`: Prints a message "---- FREE MEMBER ----" and general member information.

Add subclass method `getNumPrivateRepo()`: Returns number of private repositories

**ProMember class** (This class extends Member class)

**Overloading** the constructor method to receive a required parameter (fee)

**Overriding** printMemberInfo(): Prints a message "---- PRO MEMBER ----" and general information.

### Expected output

```
Task 1: Adding new repositories into free member account
Name: lab7 (PRIVATE) cannot be added because the number of private repository is reaching the limit
Name: lab9 (PRIVATE) cannot be added because the number of private repository is reaching the limit
*****
Task 2: Remove/add a repository from free account
Name: lab1 (PRIVATE) is successfully removed.
*****
Task 3: Display free member information
---- FREE MEMBER ----
Email: abc@mu.edu (pwd: 1234)
List of repositories
Name: lab0 (PUBLIC)
Name: lab2 (PUBLIC)
Name: lab3 (PRIVATE)
Name: lab4 (PUBLIC)
Name: lab5 (PRIVATE)
Name: lab6 (PUBLIC)
Name: lab8 (PUBLIC)
Name: lab7 (PRIVATE)
*****
Task 4: Add/remove new repositories into pro member account
Name: lab2 (PUBLIC) is successfully removed.
*****
Task 5: Display pro member information
---- PRO MEMBER ----
Member fee: 200.0
Email: xyz@mu.edu (pwd: 9876)
List of repositories
Name: lab0 (PUBLIC)
Name: lab1 (PRIVATE)
Name: lab2 (PUBLIC)
Name: lab3 (PRIVATE)
Name: lab4 (PUBLIC)
Name: lab5 (PRIVATE)
Name: lab6 (PUBLIC)
Name: lab7 (PRIVATE)
Name: lab8 (PUBLIC)
Name: lab9 (PRIVATE)
*****
```

#### Peer Bonus (Optional):

Sticker  
Here

Sticker  
Here

Sticker  
Here

### Challenge Bonus (Optional): Modify ProMember class

Suppose each pro member can have a list of collaborators.

- Add a static field COLLABORATOR\_FEE = 80.00
- Add addCollaborator(String username): If the username is not empty, adds it to the list, prints the successful message and returns true. Otherwise, returns false.
- Add removeCollaborator(String username): If the username is found, removes it from the list and returns true. Otherwise, prints a warning message and returns false.
- Add getMonthlyBill(): The monthly bill includes the pro member fee and collaborators fee. The member has to pay 80.00 per month per collaborator.
- Override printMemberInfo(): to print pro member information and list of collaborators

ProMember
- fee: double - COLLABORATOR_FEE: double - collaborators: ArrayList<String>
+ ProMember(email: String, password: String, fee: double) + printMemberInfo(): void + addCollaborator(String username): boolean + removeCollaborator(String username): boolean + getMonthlyBill(): double

### Expected output

```
Challenge 1: Add/Remove collaborator
alan@turing is added successfully.
bill@gates is added successfully.
steve@jobs is added successfully.
mark@zuckerberg is added successfully.
steve@job cannot be removed.
steve@jobs is removed successfully.
*****
```

```
Challenge 2: Get monthly bill of pro member account
The monthly bill: 440.0
*****
```

```
Challenge 3: Display pro member info
---- PRO MEMBER ----
Member fee: 200.0
```

```
Email: xyz@mu.edu (pwd: 9876)
List of repositories
Name: lab0 (PUBLIC)
Name: lab1 (PRIVATE)
Name: lab3 (PRIVATE)
Name: lab4 (PUBLIC)
Name: lab5 (PRIVATE)
Name: lab6 (PUBLIC)
Name: lab7 (PRIVATE)
Name: lab8 (PUBLIC)
Name: lab9 (PRIVATE)
-----
List of collaborators
alan@turing, bill@gates, mark@zuckerberg,
```