

Question 1: (60 minutes, 09.30-10.30)

1. Instruction

- 1) Create Java Project named “2110215_Midterm_Q1”.
- 2) Copy all folders in “Q1.zip” to your project directory src folder.
- 3) You are to implement the following classes (detail for each class is given in section 4)
 - a) **Region** (package logic)
 - b) **Database** (package logic)
- 4) JUnit for testing is in package test.grader
- 5) To submit:
 - 5.1. go to src folder that you actually do the coding for this question.
 - 5.2. Zip this question’s src folder. Name it YOUR-ID_Q1.zip (for example, 6332112121_Q1.zip)
 - 5.3. Submit the zipped file as an assignment on MyCourseville.

2. Problem Statement: CPQuest



You are tasked to finish the backbone code for a “Quest” system for a game. The demo for the system (result from running Main in package main) is shown in section 6. But it is not necessary to know the demo to write the backbone code of the program.

3. Implementation Detail

The class package is summarized below.

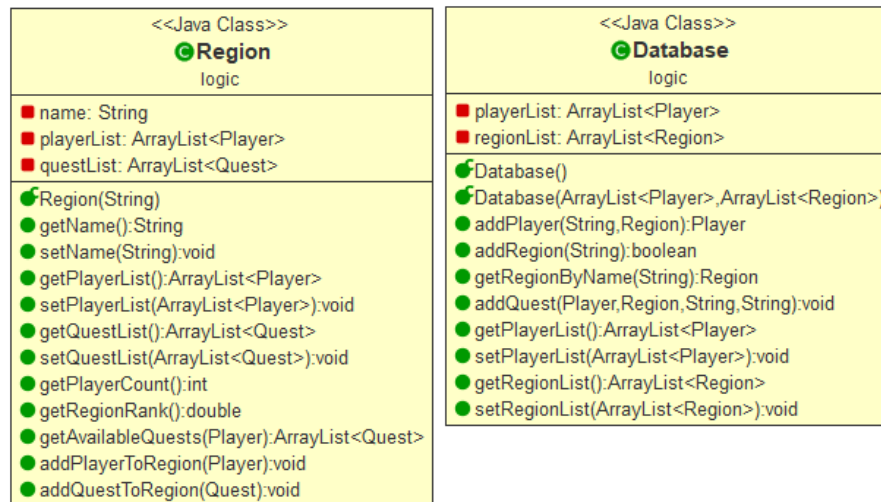


Figure 1. Class Diagram

You must write java classes using UML diagram specified above.

* In the following class description, only details of IMPORTANT fields and methods are given. *

4.1 Package logic

4.1.1 **Enum Status:** This enum contains the value that can be used to describe the status for the quest.

/*ALREADY PROVIDED*/

Values

Name	Description
AVAILABLE	This quest is available for everyone to take.
ACTIVE	This quest has been taken by someone and is in progress.
FINISHED	This quest has been finished.

4.1.2 **Class Quest:** This class represents the quest object, which will be used by Player and Region later. It contains the information about single quest.

/*ALREADY PROVIDED*/

Field

Name	Description
- Player author	The author of the quest.
- Region region	The region the quest takes place in (can be different than the author's home region)
- String name	The name of the quest.
- String description	The description of the quest.
- Status status	The current status of the quest.
- int rank	The rank of the quest.

Constructor

Name	Description
+ Quest(Player author, Region region, String name, String description)	Create a new Quest with the specified informations. Set the status to Status.AVAILABLE, and the rank to the author's current rank.

Method

Name	Description
+ getter/setter for each field	

4.1.3 **Class Player:** This class represents a player. It contains player information and status.

/*ALREADY PROVIDED*/

Field

Name	Description
- String name	The name of the player.
- int score	The score of the player
- Quest currentQuest	The player's current ongoing quest. This can be null if the player does not have ongoing quest.

Constructor

Name	Description
+ Player(String name)	Create a new Player with the specified name. Set the score to 0 and currentQuest to null.

Method

Name	Description
+ void setName(String name)	Set the name of the Player.
+ void setScore(int score)	Set the score.
+ void addScore(int amount)	Adds the specified amount to the score.
+ int getRank()	Returns the rank of the player depending on their score. The rank is - 4 (score > 10000) - 3 (7500 < score <= 10000) - 2 (5000 < score <= 7500) - 1 (2500 < score <= 5000) - 0 (score <= 2500)

+ remaining getter/setter for each field	
--	--

4.1.4 **Class Region:** This class represent a single region. It contains all data related to the region, including the quest list within the region.

You must create this class from scratch.

Field

Name	Description
- String name	The name of the Region.
- ArrayList<Player> playerList	An ArrayList containing all Player in this Region.
- ArrayList<Quest> questList	An ArrayList containing all Quest in this Region.

Constructor

Name	Description
+ Region(String name)	Create a Region with the specified name. Initialized the ArrayList for both playerList and questList

Method

Name	Description
+ void setName(String name)	Set the name of the Region. If the name is blank (Can be checked using name.isBlank()), set it to "Nowhere"
+ int getPlayerCount()	Return the size of the playerList
+ double getRegionRank()	Return the average rank of all players in the region. (Continued next page)

	<p>The average rank can be obtained by summation of the rank of every player in the region, then divide by the total numbers of the player in the region.</p> <p><u>Please round the number to 2 decimal points</u></p>
+ ArrayList<Quest> getAvailableQuests(Player viewer)	<p>Return an ArrayList of all Quest in the region that have the status AVAILABLE.</p> <p>Do note that the viewer must not be able to view the quest that belongs to himself/herself.</p>
+ void addPlayerToRegion(Player p)	Add Player to the playerList
+ void addQuestToRegion(Quest q)	Add Quest to the questList
+ remaining getter/setter for each field	

4.1.5 **Class DatabaseUtil:** This class represents the utility function that has been partially implemented. It contains useful functions you can use to help implementing the Database.

/*ALREADY PROVIDED*/

Method

Name	Description
<u>+ boolean isPlayerExists(ArrayList<Player> playerList, String name)</u>	Return true if the Player with the given name already exists in the playerList.
<u>+ boolean isRegionExists(ArrayList<Region> regionList, String name)</u>	Return true if the Region with the given name already exists in the regionList.

4.1.6 **Class Database:** This class represents the database. It contains all the frontend-backend communications, as well as the players and region list.

*For simplicity reasons for the Exam, we have used standard Exception here. In real-life scenario, please create a more specific exception for each scenario. *

You must create this class from scratch.

Field

Name	Description
- ArrayList<Player> playerList	An ArrayList of Player.
- ArrayList<Region> regionList	An ArrayList of Region.

Constructor

Name	Description
+ Database()	Create a new Database object. Initialize playerList and regionList with empty ArrayList
+ Database(ArrayList<Player> playerList, ArrayList<Region> regionList)	Create a new Database object. Initialize playerList and regionList with the specified ArrayList

Method

Name	Description
+ Player addPlayer(String name, Region region) throws Exception	<ul style="list-style-type: none">● If the player with the given name does not exist, create a new Player object with the specified detail, then add it to the playerList. <u>Don't forget to add the player into the region's own player list using addPlayerToRegion from Region class.</u>● Otherwise, throw an Exception. <p>This method returns the newly created Player object.</p>

+ boolean addRegion(String name)	If the region with the given name does not exist before, create a new Region object with specified detail, then add it to the list and return true . Otherwise, return false .
+ Region getRegionByName(String name)	Return the Region object from the regionList with the specified name. If no region with the specified name exists, then return null.
+ void addQuest(Player author, Region region, String name, String description)	Create the Quest object with the specified detail, then add it into the specified region properly. <u>Hint:</u> Use addQuestToRegion from Region class.
+ getter/setter for each field	

4.2 Package main

4.2.1 Class Main: This class is the main application. It contains methods required to run the app, as well as the main method. You can run this class to test the application.

/*ALREADY PROVIDED */

4. Score Criteria

The maximum score for the problem is 20 and will be rounded down to 5.

5.1 Class Region (RegionTest): = 12 points

testConstructor	= 1 points
testConstructorEmpty	= 1 point
testSetName	= 1 points
testSetNameEmpty	= 1 points
testAddPlayerToRegion	= 1 points
testAddQuestToRegion	= 1 points
testGetPlayerCount	= 1 points

testGetRegionRank = 2 points

testGetAvailableQuests = 3 points

5.2 Class Database (DatabaseTest): = 8 points

testAddPlayer = 1 points

testAddPlayerRepeat = 1 points

testAddRegion = 1 points

testAddRegionRepeat = 1 points

testAddQuest = 2 points

testGetPlayerList = 1 points

testGetRegionList = 1 points

5. Program Demonstration

```
Quest management system.
=====
Welcome, Guest!
What are you going to do?
=====
1) Log In
2) View Statistics
3) Exit Program
=====
>>
```

Figure 2: The Initial Screen

First off, as a guest. You can either log in or viewing the statistic. Once you logged in, you will have more options to work with. We will cover the statistics option later.

Once you picked the log in option, you will be presented with the current user list in the system, or you can choose to make the new one. Do note that the new username cannot be the same as the one in the database. If it happened, the system would alert the user.

```

=====
Please pick the Player Log In.
=====
1) Chrono
2) Tokoha
3) Shion
4) Jaime
5) Register New Player
=====
>> |

```

```

>> 5
Enter the new Player Name
>> Chrono
Enter the Region name.
>> Asia
Could not create the player: The player with the name Chrono already exists.
=====

```

Figure 3: Registering new player

Once you logged in, the start menu changed, and you can access more option.

```

Welcome, Chrono!
Your point is 0, RANK 0
You don't have the ongoing quest.
What are you going to do?
=====
1) Edit Name
2) Take On a Quest
3) Post Quest
4) View Statistics
5) Log Out
=====
>>

```

Figure 4: New Start Menu once logged in

For the first option, you can edit your own username, do note that the same restriction as when registering new player applies.

```

>> 1
Enter the new name (Please leave it empty if unedited)
>> Tokoha
Could not rename the player: The other player with the name Tokoha already exists.
=====

```

Figure 5. Submenu for editing username

The next option is “Take On a Quest”. You can take on a quest using this option. The system will prompt you to pick the region and will display the current available quest in that region.

```
>> 2
=====
Which Region would you like to take on the quest?
=====
1) Asia
2) Africa
3) Australia
4) Europe
5) North America
6) South America
=====
>> 1
=====
Select the quest you would like to take
=====
1) I WANT TO EAT
2) Urgent Help Needed
=====
>> 1
=====
Quest Detail
I WANT TO EAT
Location: Asia Rank: 0
Author: Jaime
Can you help me find somewhere to eat?
=====
Do you accept this quest? (Y/N)
=====
>> y
You accept the quest!
```

Figure 6. Take On a Quest

Do note that this option will be changed to Manage Quest Status instead when you have taken a quest. With Manage Quest Status menu, you can mark the quest as finished or decline the quest. By marking the quest as finished, you will obtain the score reward from the quest equal to the quest’s rank.

```

Welcome, Chrono!
Your point is 0, RANK 0
You have the current ongoing quest: I WANT TO EAT
What are you going to do?
=====
1) Edit Name
2) Manage Quest Status
3) Post Quest
4) View Statistics
5) Log Out
=====
>> 2
Your Current ongoing quest is: I WANT TO EAT
What are you going to do?
=====
1) Mark Current Quest as Finished
2) Decline Current Quest
3) Back
=====
>>

```

Figure 7. Manage Quest Status

The third option is for posting your own quest. Everything here should be self-explanatory.

```

>> 3
Enter the Quest Name
>> Quest Title
Enter the Quest Description
>> Quest Description
=====
Please pick the Quest's Region
1) Asia
2) Africa
3) Australia
4) Europe
5) North America
6) South America
=====
>> 1
|=====

```

Figure 8. Posting Quest Option

Finally, the last option is for viewing the status of the region. It displays the player count and average rank of the players in the region.

```
=====
>> 4
=====
Please pick Region you would like to view.
1) Asia
2) Africa
3) Australia
4) Europe
5) North America
6) South America
=====
>> 1
Region name: Asia|
Player Count: 3 Average Rank: 2.0
=====
```

Figure 9. Viewing Region Status