OOP Project Question 3 Code Evaluation

Quizzy–The Online Quiz System

By

Ishaan Sarna

2019A8PS0388P

f20190388@pilani.bits-pilani.ac.in

Originally, there were two people but my partner has dropped the course

GitHub link

GitHub - ishaansarna/OOP-Project-Question-3-Online-Quiz-System: Project for OOP CS F213,

Project for OOP CS F213, odd semester, AY 21-22, BITS Pilani, Pilani Campus - GitHub - ishaansarna/OOP-Project-Question-...

https://github.com/ishaansarna/OOP-Project-Question-3-Online-Quiz-System

Demonstration video link

https://youtu.be/sw6431a2kj4

Description of the classes

1. Main

The Main class is the starting point of the program. It contains only the main method which creates a new Admin class and then starts the thread of the created instance.

2. Admin

The admin class is used to create and work with an admin user. It implements the **Runnable interface** and contains a Thread data member.

Admin class stores lists of **Question**s and **Student**s that are part of the quiz.

Methods

1. addDefaultSetOfStudentsAndQuestions(): Add a default set of students and questions for the quiz.
2. addNQuestions(): Take input n from the admin and then let them add n questions to the question bank.
3. addNStudents(): Take input n from the admin and then let them add n students to the student list.

4. addQuestion(): Let the user add a new question.
5. addStudent(): Let the user add a new student.
6. conductQuiz(): Start the quiz. The students answer the questions in a round-robin fashion. After the quiz the students can view their scores and leaderboard position along with the details of their attempts.
7. displayLeaderboard(): The admin can view the leaderboard with the students arranged in the order of their scores.
8. exit (): Quit the program.
9. removeQuestion(): The method takes a string input from the admin and then removes the question from the question bank.
10. removeStudent(): The method takes a string input from the admin and then removes the student from the students list.
11. resetStudentsOrder(): This is an internally used method for sorting students by their roll number after sorting them with their leaderboard position.
12. run(): This is the driver method for the program which is called when the admin thread is started in the main method. It runs an infinite loop until the admin exits the program. Inside the loop, a switch case runs that lets the admin choose from:-
    1. Conduct quiz
    2. Add a student
    3. Add n students
    4. Add a question
    5. Add n questions
    6. View all students
    7. View all questions
    8. Remove a student
    9. Remove a question
    10. View the leaderboard
    11. Add default set of students and questions
    12. Exit
13. sortStudentsByLeaderboardPosition(): This method sorts the students by their score for displaying the leaderboard.
14. viewQuestions(): This method lets the admin view the questions.
15. viewStudents(): This method lets the admin view the details of all the students.

3. CompareByScore

This is a utility class that implements Comparator<Student> for comparing students by their total scores.

Method

1. compare(student, student): Returns an integer after comparing the two students by their score.

4. CompareByRollNo

This is a utility class that implements Comparator<Student> for comparing students by their roll numbers.

Method

1. compare(student, student): Returns an integer after comparing the two students by their roll number.

5. Student

This class is used to create students and implements the **Runnable interface**. It **contains a thread property allowing each student to work in their own thread.**

Methods

1. attemptQuestion(Question): Internal method to attempt the allotted question during a quiz.
2. createStudentPrompt(Admin): Allows the admin to add a student interactively.
3. createThread(): Creates a thread for the student.
4. displayAttemptedQuestions(): Allows the students to view their attempts after the quiz.
5. run(): The driver for the student thread as required by the Runnable interface. It allows the student to attempt a question when the quiz is running and view their performance and attempts after the quiz.
6. runThis(): This is the primary method to connect the student to the admin and abstract the details about the inner workings of the student's thread away.
7. showBasicDetails(): Allows the student to view their performance (score and leaderboard position).
8. showDetails(): Shows the basic details and then allows the student to view their attempts.
9. updateScore(int): Utility method to add points to the total score after a question.
10. viewStudent(): Allows the admin to view the details of the student.

6. Question

This class is used as a container for a question, its options and all the related methods.

It contains member variables for the question string, options list, correct answer, number of points, time allowed and the option selected by the playing student.

Methods

1. addQuestion(): Static method to interactively create a question and return its instance to add to the question bank.

2. attemptQuestion(): Method to let the student attempt a question during the quiz. It shows the questions along with the options, takes the input from the student through the Timer class and then tells whether the student answered correctly. It returns the number of points scored in the question.
3. returnAttemptedQuestion(): Returns the attempted question to add to the student's attempted questions list for post-quiz review
4. viewQuestion(): To allow the admin to view all the information about the question.

7. AttemptedQuestion

This is a container class to store the details about all the questions that have been attempted by the student in a list in the student's instance. It stores the question, correct answer, chosen answer and whether the answer is correct.

All student instances store a list of attempted questions.

Method

1. display(): It displays the details of the attempted question to allow the student to do post-quiz review.

8. Timer

This class provides a utility method to allow the student to respond within a stipulated time barring which the question would timeout.

Method

1. quizTimedQuestion(long, long): method to return the inputted integer if the input was made in the stipulated time.

Known limitations and bugs

The students are limited to using a single device and therefore are compelled to play in a round-robin fashion.

Currently, a GUI has not been implemented in the program.

There are no known bugs as of release 1.0 on GitHub

Analysis of design patterns

Model-View-Controller design pattern

The MVC design pattern separates the program into 3 portions–model, view and controller.

1. Model - responsible for representing the data. It may contain any custom classes, databases etc.

2. View - responsible for conveying information to the user. Takes care of the UI.
3. Controller - responsible for controlling the interactions between the model and the view.

Currently, the code is partitioned by what classes it works with. For example the model, view and controller logics of Question and Student classes reside inside those classes and are not segregated.

Another way to write the program would have been to separate the model, view and controller logic into different classes. This means that the model would not have to worry about how the data is shown to the user. If this were the case, implementing a GUI would have involved only working with the view portion of the program, leaving model and controller alone. Currently, shifting to a GUI would require more fundamental changes as the model and view are more tightly coupled than if MVC pattern was implemented.

Iterator design pattern

Currently, there is no good way to segregate the questions in the program for a quiz in case we want to do something like asking a different set of questions based on difficulty, topic etc.

If we would have implemented the iterator pattern, then we could have easily created different sets of questions which would all have implemented the Iterator interface. The admin could have then easily customised the quiz without worrying about how the different sets are stored.

Factory design pattern

Currently, the Question class is hardcoded to work for multiple choice, single correct questions. Instead, we could have created an interface that just provided a structure for questions. Then it would have been up to the implementation to figure out the internal workings like taking an input, checking the correctness of the answer etc.

A factory class could have been created so that the admin could easily create any type of question and add them to the same list as all of them would have the same reference type.

Analysis of the principles of object oriented programming

Encapsulate what varies

The code heavily applies the principle of encapsulating the portions that might change. The changes may be due to change in the requirements such as adding new features or modifying existing ones.

An example in the code can be seen in the view methods for students and questions. If, for some reason like adding a GUI, we want to change how displaying students or questions work, we can just modify the methods and not touch the admin class which just calls those methods.

Another example can be seen in the Timer class which abstracts away the logic of taking input within a stipulated time away from the question class. This means that the question, student and admin class do not need to worry about the timer logic.

Favour composition over inheritance

Inheritance is in most cases, a rigid concept as java does not allow inheriting from more than one class at the same time. It is required when we want to use Class B where Class A might be expected (here Class B inherits from Class A). While, when applying composition, we can selectively access behaviours of a class from another.

For example, a dog requires all behaviour of an animal if not more, so it is appropriate for a dog to inherit from an animal class. While as both a dog and a car might need movement behaviour, extracting it into a interface will be beneficial.

An example in the program can be seen with the CompareByRollNo and CompareByScore classes. As both of them have the compare behaviour for the student class, they implement the Comparator<Student> interface.

Strive for loose coupling between objects that interact

Loose coupling means that the two objects need not have too much information about each other. This principle is used throughout the program as all the behaviour relating to certain classes like Student or Question has been put in methods inside the class. Which means that Student does not need to worry about how to attempt a question as this behaviour is inside the Question class. In the same way Student need not worry about how Admin creates the leaderboard. Thus, all the nitty gritty about how to do something is inside the relevant classes and the other classes just need to call the appropriate methods like `quizTimedQuestion(long, long)`, `attemptQuestion()` and `updateScore()`.

Classes should be open for extension and closed for modification

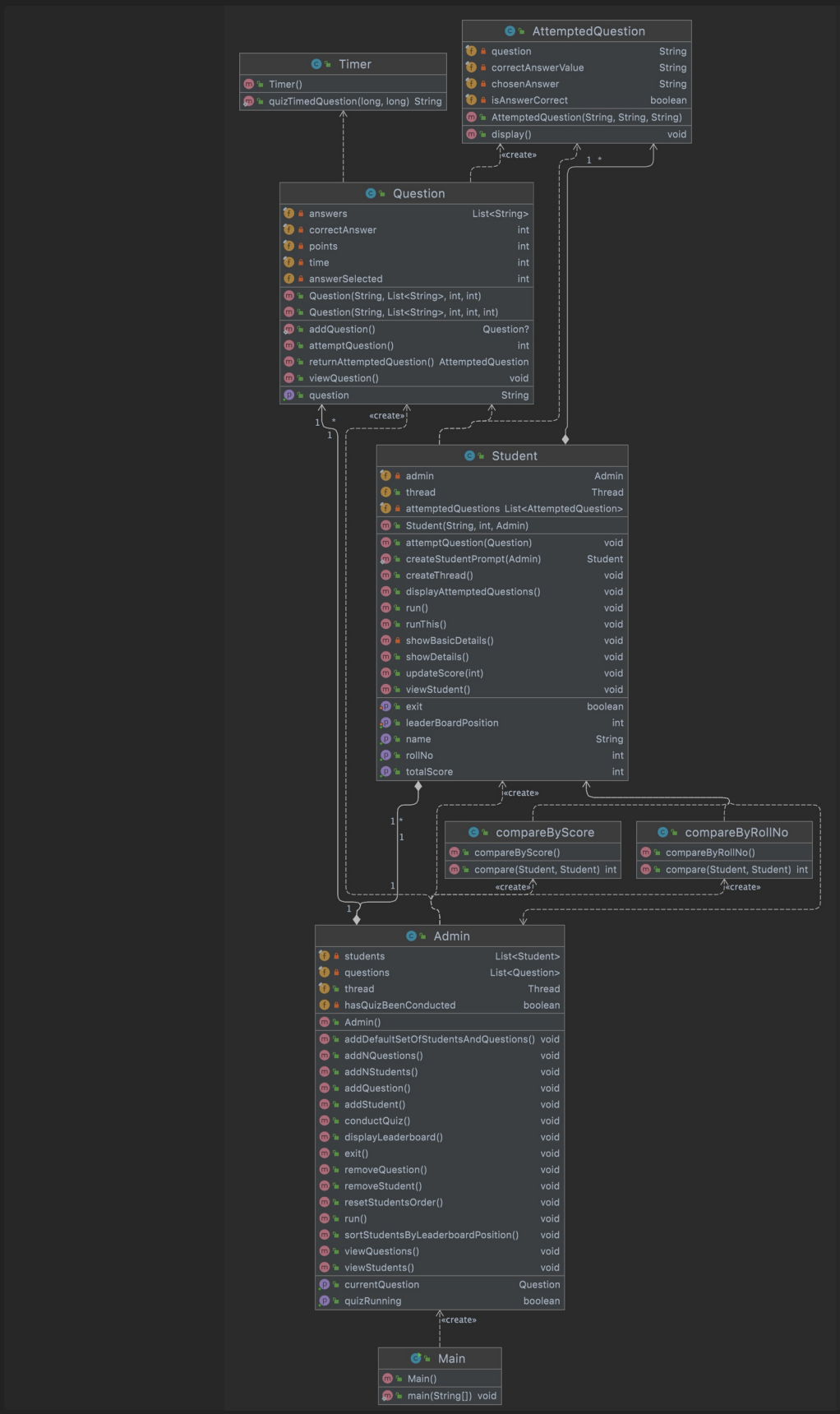
The current code works correctly and no unfixed bugs have been detected, so it is efficient to leave this code alone and extend it if any changes need to be made.

For example if we later want to add a student who just wants to audit the quiz we can extend student to create a new class that implements this behaviour while keeping the original Student class intact. Or if we want to add a new type of question with negative marks for the wrong answer we can just extend Question with an overwritten method that modifies the number of points returned on attempting the question.

Future work possible

- Implementing a graphical user interface
- Enabling online access to allow remote play
- Storing the students and questions in an SQL database instead of java lists

UML Class Diagram



Walkthrough of inputs and outputs

The program begins by creating an admin and presenting them with the options available. The admin can then enter their choice.

```
Welcome to Quizzy-The Online Quiz System
----------------------------------------------------------------
What would you like to do? Please select the appropriate option.
1. Conduct quiz
2. Add a student
3. Add n students
4. Add a question
5. Add n questions
6. View all students
7. View all questions
8. Remove a student
9. Remove a question
10. View the leaderboard
11. Add default set of students and questions
12. Exit
```

The admin can now add questions and students

Adding students

```
3
How many students would you like to add
3
Please add student number 1
Please enter the name of the student
Jacob
Please enter the roll number of the student
10
Please add student number 2
Please enter the name of the student
Rachel
Please enter the roll number of the student
11
Please add student number 3
Please enter the name of the student
Tommy
Please enter the roll number of the student
12
```

Adding questions

```
5
How many questions would you like to add
6
Please add question number 1
What is the question
Which is the largest planet?
What are the options? (Enter an option on each line, enter "d" to
Saturn
Uranus
Jupiter
d
Please enter the option number of the correct answer
3
Please enter the number of points
1
Please enter the time for this question (enter 0 for default valu
0
```

```
What is the question
What is the name of the Vice President of India?
What are the options? (Enter an option on each line, enter "d" to finish adding)
Narendra Modi
Ramnath Kovind
Venkaiah Naidu
d
Please enter the option number of the correct answer
3
Please enter the number of points
1
Please enter the time for this question (enter 0 for default value)
15
```

Similarly other 4 questions are added.

Removing a student

```
8
Enter the name of the student to be removed
Mary
Successfully removed Mary
```

Removing a question

```
Enter the question to be removed
Who is the Prime Minister of the United Kingdom?
Successfully removed the question "Who is the Prime Minister of the United Kingdom?"
```

Viewing all students

```
6
The students are:-
1)
Name: Jacob
Roll No.: 10
Total Score: 0
Did not take part in a quiz yet so leaderboard position is not applicable
2)
Name: Rachel
Roll No.: 11
Total Score: 0
Did not take part in a quiz yet so leaderboard position is not applicable
3)
Name: Tommy
Roll No.: 12
Total Score: 0
Did not take part in a quiz yet so leaderboard position is not applicable
4)
Name: Christine
Roll No.: 21
Total Score: 0
Did not take part in a quiz yet so leaderboard position is not applicable
```

Viewing all questions

```
7
The questions are:-
1)
The question is -
Which is the largest planet?
The options are -
1) Saturn
2) Uranus
3) Jupiter
Out of which the correct option is "Jupiter"
2)
The question is -
What is the name of the Vice President of India?
The options are -
1) Narendra Modi
2) Ramnath Kovind
3) Venkaiah Naidu
Out of which the correct option is "Venkaiah Naidu"
3)
The question is -
Which is the third most populous country?
The options are -
1) India
2) United States of America
3) Indonesia
4) Russia
Out of which the correct option is "United States of America"
```

```
4)
The question is -
Who directed, produced, executive produced, starred in and wrote the film 'The Room'?
The options are -
1) Tommy Wiseau
2) Charlie Sheen
3) Jared Leto
4) Alec Baldwin
5) Bruce Willis
Out of which the correct option is "Tommy Wiseau"
5)
The question is -
BITS Pilani had to shut down due to student strikes in 1973, 1980 ans ____?
The options are -
1) 2001
2) 1985
3) 1991
Out of which the correct option is "1985"
6)
The question is -
Who was one of the founders of Wikipedia?
The options are -
1) Jimmy Wales
2) Steve Jobs
3) Steve Wozniak
4) Larry Page
Out of which the correct option is "Jimmy Wales"
```

Conducting the quiz

This quiz has been conducted with the sample set of students and questions available in the program.

1. **Getting a question correct**

```
1
Question for John with roll no. 1
For 2 points with a time limit of 20 seconds
---------------
Fathometer is used to measure?
---------------
1) Earthquakes
2) Rainfall
3) Ocean depth
3
Congratulations, you have got 2 points!
Your current score is 2
```

2. Getting a question incorrect

```
------------------------
Question for Amy with roll no. 2
For 3 points with a time limit of 30 seconds
---------------
Epsom(England) is the place associated with
---------------
1) Snooker
2) Horse racing
3) Shooting
3
You got it wrong! Better luck next time ;)
Your current score is 0
```

3. Timing out on a question

```
------------------------
Question for Lisa with roll no. 3
For 1 points with a time limit of 10 seconds
---------------
Which is the largest organ in the human body?
---------------
1) Heart
2) Skin
3) Large Intestine
You ran out of time :(
Your current score is 0
```

Similarly all other questions are attempted and the students are given an option to review their performance.

```
John, do you want to see your details? [y/n]
y
Name: John
Roll No.: 1
Total Score: 5
Leaderboard Position: 1
Would you like to see your attempts? (y/n)
y
You got the question "Fathometer is used to measure?" correct!
The answer was Ocean depth

You got the question "What is the French word for 'hat'?" correct!
The answer was Chapeau

You got the question "Where is Tasmania located?" correct!
The answer was Australia

Thank you for playing

Amy, do you want to see your details? [y/n]
y
Name: Amy
Roll No.: 2
Total Score: 5
Leaderboard Position: 2
Would you like to see your attempts? (y/n)
n
Thank you for playing

Lisa, do you want to see your details? [y/n]
n
Thank you for playing
```

Finally, the leaderboard is shown

```
The Leaderboard
----------------------------------------
Rank 1 is John with a score of 5
Rank 2 is Amy with a score of 5
Rank 3 is Lisa with a score of 4
Rank 4 is Mark with a score of 4
------------------------------------------------------------
```

Exiting

```
12
Thank you for  using Quizzy-The Online Quiz System

Process finished with exit code 0
```