

Specifications

Problem Description

You are going to write a class that stores the data for a trivia question to be used in a provided trivia quiz program.

Input: The class will obtain no inputs from an external source such as the user or a file. The trivia quiz program (provided for you) will interact with the class through its public interface.

Output: The class will produce no outputs to an external source such as standard output or a file.

Interface: There is no user interface to the class, only a public interface to other program components, detailed in the design specifications section.

Design specifications:

TriviaQuiz.java

TriviaQuiz is written for you. It creates three TriviaQuestion objects – two using hard-coded values, and one using the default constructor. It then poses the questions, obtains answers from the user, and reports the results. Do not modify this program in any way.

The program will work correctly with the class, if the class is created to specifications. You may wish to write additional testing programs to test your class.

Note that TriviaQuiz.java uses two helper classes (ConsoleInput.java and ConsoleInputReadTask.java) so time hints can be provided to the user. I included this feature to make testing your TriviaQuestion.java class easier. You can ignore these classes.

TriviaQuestion.java

Objects of this class will represent a single trivia question. A trivia question might look like:

```
Who was Alan Turing?  
1) first-team midfielder for Manchester United  
2) the father of theoretical computer science  
3) the true name of the Red Baron  
4) 19th century Scottish playwright
```

A trivia question has a right answer and is worth some number of points.

There are two versions of this class that you can write – the basic version and the enhanced version. Start by writing the basic version, and then challenge yourself by attempting the enhanced version. Writing the enhanced version will help you if you move on to Advanced Java and it will give you good practice with the important skill of working with a Java library.

Basic version

A TriviaQuestion object will have the following fields: `int correctAnswer`, `points`, `String question`, `answer1`, `answer2`, `answer3`, `answer4`, `boolean posed`.

`question` will be the question text (e.g. "Who was Alan Turing? "). `answer1` through `answer4` are the four possible answers (e.g. "first-team midfielder for Manchester United," "the father of theoretical computer science," etc.). `correctAnswer` is the integer value for the correct answer (e.g. 2, if `answer2` is the correct answer). `points` is the number of possible points awarded for answering the question correctly. `posed` is initially set to false. A question can be created without being posed. Once the question is posed, the field is set to true.

A `TriviaQuestion` object will have the following methods:

- **Constructor (with arguments)**
The full argument constructor will take arguments for the `question`, `answer1`, `answer2`, `answer3`, `answer4`, `correctAnswer`, and `points` fields. `posed` should be initialized to false. It is not a parameter.
- **Constructor (no arg)**
The no-arg constructor will instantiate the `question`, `answer1`, `answer2`, `answer3`, `answer4`, `correctAnswer`, and `points` fields to something bland and suitable for testing, such as "What is the first letter of the alphabet? ", "J", "Z", "M", "A", 4, 10. As with the full argument constructor, `posed` should be initialized to false.
- **getScore accessor**
The `getScore` accessor will take an integer parameter representing an answer and will return a double score based on whether the answer is correct. If the question hasn't been posed, the `getScore` method should return 0.
- **poseQuestion accessor**
The `poseQuestion` accessor will set the `posed` field to true and return a String composed of the question and list of answers.
- **Mutators**
Write mutators for the `question`, `answer` (`answer1`, `answer2`, etc.), and `points` fields. The mutators will change the values in the fields to the parameter passed *only if the question hasn't already been posed*. Once a question has been posed, it cannot be changed.

Hints: Make sure you understand the specifications of this class so you can write it correctly, and ask questions! You should create a class diagram and discuss it in the discussion board before proceeding with implementation.

Enhanced Version

The enhanced `TriviaQuestion` class will have a timing component. You will implement this by doing the following:

Add the following constants, representing the number of seconds within which the question must be answered in order to receive the named percentage of total points:

HUNDRED (10), NINETY (15), SEVENTYFIVE (20), FIFTY (25), THIRTY (30), TEN (35).

In other words, if the question is answered correctly within 10 seconds of being posed, the user receives 100% of the question points. If the question is answered correctly between 10 and 15 (inclusive)

seconds, then the user receives 90% of the question points, etc. If the user takes longer than 35 seconds to answer the question, even if answered correctly, they receive no points.

Add the field: `Instant start`.

`start` is of class `Instant`, which is a Java class that allows you to do calculations with time. You will want to use `Instant` to determine how much time has passed between when the question was posed and when the question was answered. See the **Appendix** at the end of this assignment for information on this class and how to use it in your program.

Once a question is posed, the timer starts. That is when the `Instant` object `start` should be instantiated – when the question is posed.

Modify the methods as follows:

- Constructor (with arguments / no arg)
`start` will be initialized to `null`.
- `getScore` accessor
The `getScore` accessor will take an integer parameter representing an answer, and will return a double score based on whether the answer is correct and how much time has elapsed. (For example, if more than 10 seconds have elapsed, but less than 15, then the score will be 90% of the `points` field, etc.) If the question hasn't been posed, the `getScore` method should return 0.
- You might optionally wish to have a private method as a helper method for `getScore`, such as `getPercent`, which returns the percentage of points the user will earn based on elapsed time. Having this private method will simplify the `getScore` method.
- `poseQuestion` accessor
The `poseQuestion` accessor will instantiate the `start` field with an `Instant` object representing the current time.

Hints: Spend a good amount of time working with the `Instant` class. Make sure you understand what it does and how to work with it. Feel free to discuss it in the discussion board.

Testing

Test Case 1

Purpose

Unit test case of the `TriviaQuestion` class. Write a program to unit test your class. The `TriviaQuestion` object is instantiated to the values passed in as parameters to the constructor. The `TriviaQuestion` object, once instantiated and before receiving the `poseQuestion` method, allows the test program to change all fields. Send the object all mutator messages in your unit test program. The `poseQuestion` method returns a correct string (display to the console). After the question is posed, the object will not allow the test program to change any fields. Send the object all mutator messages in your unit test program. The object responds with full points for a question answered correctly (within ten seconds of being posed, for the Enhanced Version of the class), and responds with zero points for a question answered incorrectly.

Input

Instantiate a `TriviaQuestion` object with the following parameters to the constructor:

Question: "Which 2016 release starred Gal Gadot?"

Answer 1: "The Last Yoda"

Answer 2: "Batgirl"

Answer 3: "Running Man 2049"

Answer 4: "Ender's Game"

Correct answer: 2

Points: 5

Change each field as follows:

Question: "Which 2017 release starred Gal Gadot?"

Answer 1: "The Last Jedi"

Answer 2: "Wonder Woman"

Answer 3: "Blade Runner 2049"

Answer 4: "Molly's Game"

Points: 10

Expected output

The object before posing and attempting to mutate fields:

(x)= Variables Breakpoints Expressions	
Name	Value
no method return value	
args	String[0] (id=20)
question	TriviaQuestion (id=21)
answer1	"The Last Yoda" (id=30)
answer2	"Batgirl" (id=36)
answer3	"Running Man 2049" (id=37)
answer4	"Ender's Game" (id=38)
correctAnswer	2
points	5
posed	false
question	"Which 2016 release starred Gal Gado..."
start	null
keyboard	Scanner (id=24)

The program running in the console:

Put a break point here.

Attempting to change all mutable question fields before posing.

Which 2017 release starred Gal Gadot?

1) The Last Jedi

2) Wonder Woman

3) Blade Runner 2049

4) Molly's Game

Please enter the correct answer (2): 2

You scored 10.0 points.

Which 2017 release starred Gal Gadot?

- 1) The Last Jedi
- 2) Wonder Woman
- 3) Blade Runner 2049
- 4) Molly's Game

Please enter an incorrect answer (anything except 2): 1
You scored 0.0 points.

Attempting to change all mutable question fields after posing.

The object after posing and attempting to mutate fields:

question	TriviaQuestion (id=21)
answer1	"The Last Jedi" (id=40)
answer2	"Wonder Woman" (id=41)
answer3	"Blade Runner 2049" (id=42)
answer4	"Molly's Game" (id=43)
correctAnswer	2
points	10
posed	true
question	"Which 2017 release starred Gal Gadot? ...
start	Instant (id=45)

Test Case 2

Purpose

Run TriviaQuiz.java. If you have written the *Enhanced Version* of the class, answer the questions correctly but timed so they fall into the 90, 75, and 50% categories. Run the program a second time and time your correct answers to fall into the 30 and 10% categories, and then answer the final question after the complete time has passed (so that you get zero points). If you did not write the *Enhanced Version*, answer the first two questions correctly and the last question incorrectly.

Input

10-15 second delay, 4
15-20 second delay, 2
20-25 second delay, 2

25-30 second delay, 4
30-35 second delay, 2
> 35 second delay, 2

Expected output

What is the first letter of the alphabet?

- 1) J
- 2) Z
- 3) M
- 4) A

Please enter the number of your answer: 5 seconds have passed...
Please enter the number of your answer: 10 seconds have passed...
Please enter the number of your answer: 4
You scored 9.0 points.

Who was Alan Turing?

- 1) first-team midfielder for Manchester United
- 2) the father of theoretical computer science
- 3) the true name of the Red Baron
- 4) 19th century Scottish playwright

Please enter the number of your answer: 5 seconds have passed...

Please enter the number of your answer: 10 seconds have passed...

Please enter the number of your answer: 15 seconds have passed...

Please enter the number of your answer: 2

You scored 7.5 points.

Which 2017 release starred Gal Gadot?

- 1) The Last Jedi
- 2) Wonder Woman
- 3) Blade Runner 2049
- 4) Molly's Game

Please enter the number of your answer: 5 seconds have passed...

Please enter the number of your answer: 10 seconds have passed...

Please enter the number of your answer: 15 seconds have passed...

Please enter the number of your answer: 20 seconds have passed...

Please enter the number of your answer: 2

You scored 5.0 points.

What is the first letter of the alphabet?

- 1) J
- 2) Z
- 3) M
- 4) A

Please enter the number of your answer: 5 seconds have passed...

Please enter the number of your answer: 10 seconds have passed...

Please enter the number of your answer: 15 seconds have passed...

Please enter the number of your answer: 20 seconds have passed...

Please enter the number of your answer: 25 seconds have passed...

Please enter the number of your answer: 4

You scored 3.0 points.

Who was Alan Turing?

- 1) first-team midfielder for Manchester United
- 2) the father of theoretical computer science
- 3) the true name of the Red Baron
- 4) 19th century Scottish playwright

Please enter the number of your answer: 5 seconds have passed...

Please enter the number of your answer: 10 seconds have passed...

Please enter the number of your answer: 15 seconds have passed...

Please enter the number of your answer: 20 seconds have passed...

Please enter the number of your answer: 25 seconds have passed...

Please enter the number of your answer: 30 seconds have passed...

Please enter the number of your answer: 2

You scored 1.0 points.

Which 2017 release starred Gal Gadot?

- 1) The Last Jedi
- 2) Wonder Woman
- 3) Blade Runner 2049
- 4) Molly's Game

Please enter the number of your answer: 5 seconds have passed...

Please enter the number of your answer: 10 seconds have passed...
Please enter the number of your answer: 15 seconds have passed...
Please enter the number of your answer: 20 seconds have passed...
Please enter the number of your answer: 25 seconds have passed...
Please enter the number of your answer: 30 seconds have passed...
Please enter the number of your answer: 35 seconds have passed...
Please enter the number of your answer: 2
You scored 0.0 points.

Rubric

An exceptional-quality assignment will meet the following standards:

- Meeting functional and design specifications
The Java class works and meets all of the specifications, with no additional unspecified functionality. The programmer has used programming techniques from the first five lessons only. If the class misses specifications or does not function correctly, errors are acknowledged with a thorough and reflective analysis in the testing section (points will be removed for missed specifications).
- Communicating with identifiers and white space
The program makes appropriate use of variables. Variables, methods, and constants are named according to convention and are named for understandability and purpose. White space, both vertical and horizontal, is correctly used for readability and meets programming conventions.
- Communicating through documentation
The Java program contains comments including the programmer's name and date. There are block comments (as many as necessary) for each distinct block of code which accurately describe what the block is accomplishing by relating the code to the problem being solved. Javadoc is included and meets the javadoc standards.
- Assumptions and Testing
Testing is thorough. If there are errors, they are described in the testing section. If there are questions, they are answered thoughtfully in the testing section. All assumptions are made explicit.

The Oracle documentation on `Instant` can be found here:

<https://docs.oracle.com/javase/8/docs/api/java/time/Instant.html>

The `Instant` class is in the `time` package. You can therefore use either one of the following import statements at the top of your program:

```
import java.time.*
import java.time.Instant
```

Within your program, assign `Instant.now()` to a variable of type `Instant` (e.g. `Instant timestamp = Instant.now();`)

You can convert an `Instant` to the number of seconds since the Java Epoch:

```
long getEpochSecond()
Gets the number of seconds from the Java epoch of 1970-01-01T00:00:00Z.
```

And you can add and subtract:

```
Instant minusSeconds(long secondsToSubtract)
Returns a copy of this instant with the specified duration in seconds subtracted.

Instant plusSeconds(long secondsToAdd)
Returns a copy of this instant with the specified duration in seconds added.
```

When converted to a `String`, you will get a date-time string such as `"2017-04-19T10:36:45.176Z"` which represents the time that the object was created. You can invoke `toString` on the object directly to obtain a `String`:

```
String toString()
A string representation of this instant using ISO-8601 representation.
```

I recommend you practice with the `Instant` class, using a simple program such as the following. Make changes to it and predict the results until you're certain you know how to use the class and the methods shown here.

```
import java.time.Instant;
import java.util.Scanner;
/**
 * Instant practice
 */
public class InstantPractice {
    public static void main(String[] args) {
        // Create an instant
        Instant now = Instant.now();

        // Delay a bit -- press enter after
        // a number of seconds has passed
        Scanner keyboard = new Scanner (System.in);
        System.out.print("Enter your name: ");
        String delay = keyboard.nextLine();
```



```
// Make another instant, now that you've delayed
Instant later = Instant.now();

// Print the two times
System.out.println(now.toString());
System.out.println(later.toString());

// Subtract the first time (as seconds)
// from the second time to get the difference
long nowSeconds = now.getEpochSecond();
Instant difference = later.minusSeconds(nowSeconds);
System.out.println(difference.getEpochSecond());

    }
}
```