

CIS 1068 Assignment 10

Test Papers Assignment

Due: ~~Friday, April 8~~ Tuesday, April 12

80 points

description

Design a class hierarchy used to represent test papers that would be given in school. It should include at least the following:

Question

Each question has:

`points`
Number of points earned for answering the question correctly.

`difficulty`
Range of difficulty ranging from the constants `MIN_DIFFICULTY` to `MAX_DIFFICULTY`.

`answerSpace`
Amount of space that should be left on the paper so that the test taker can fill in the answer. This can be represented as the number of lines on the page.

`questionText`
The text of the question, e.g., "How much wood would a woodchuck chuck if a woodchuck could chuck wood?"

For each question, we should be able to create a String representation of the question as it should be presented to the student (for example, during a test or quiz).

ObjectiveQuestion

An `ObjectiveQuestion` is a type of question which has a definitive answer. For example, the question $2+3$ has the answer 5. (A **non-objective** question would be one where there's no single correct answer, e.g., something like, ["The Civil War was neither civil nor a war. Discuss."](#))

Each `ObjectiveQuestion` has:

`points`
Number of points earned for answering the question correctly.

`difficulty`
Range of difficulty ranging from the constants `MIN_DIFFICULTY` to `MAX_DIFFICULTY`.

`answerSpace`
Amount of space that should be left on the paper so that the test taker can fill in the answer. This can be represented as the number of lines on the page.

`questionText`
The text of the question, e.g., "How much wood would a woodchuck chuck if a woodchuck could chuck wood?"

`correctAnswer`
The correct answer to the question.

For each `ObjectiveQuestion`, we should be able to create a String representation of the question as it should be presented to the student during a Test. We should also be able to create a String representation of the question which includes the correct answer, such as one that a grader might use when grading a paper.

FillInTheBlankQuestion

A fill in the blank question is a kind of question where the correct answer fills in a missing word in the provided text. For example. " _____ was the 16th US President."

Each FillInTheBlankQuestion has:

points

Number of points earned for answering the question correctly.

difficulty

Range of difficulty ranging from the constants MIN_DIFFICULTY to MAX_DIFFICULTY.

answerSpace

Amount of space that should be left on the paper so that the test taker can fill in the answer. This can be represented as the number of lines on the page.

questionText

The text of the question, e.g., "How much wood would a woodchuck chuck if a woodchuck could chuck wood?"

correctAnswer

The correct answer to the question.

For each FillInTheBlankQuestion, we should be able to create a String representation of the question as it should be presented to the student during a Test. This should include the blank space. We should also be able to create a String representation of the question which includes the correct answer, such as one that a grader might use when grading a paper.

In a test representation, this might look something like:

| _____ was the 16th US President.

and in an answer key representation, it might look something like:

| ____Abraham Lincoln____ was the 16th US President.

MultipleChoiceQuestion

A MultipleChoiceQuestion is a kind of question in which there are multiple possible solutions but only one correct solution.

Each MultipleChoiceQuestion has:

points

Number of points earned for answering the question correctly.

difficulty

Range of difficulty ranging from the constants MIN_DIFFICULTY to MAX_DIFFICULTY.

answerSpace

Amount of space that should be left on the paper so that the test taker can fill in the answer. This can be represented as the number of lines on the page. MultipleChoiceQuestions can be answered in only one line.

questionText

The text of the question, e.g., "How much wood would a woodchuck chuck if a woodchuck could chuck wood?"

possibleAnswers

A list of possible answers, only one of which is correct.

correctAnswer

The correct answer among the possibleAnswers.

For each MultipleChoiceQuestion, we should be able to create a String representation of the question as it should be presented to the student during a Test. We should also be able to create a String representation of the question which includes the correct answer, such as one that a grader might use when grading a paper.

In a test representation, this might look something like:

| Who lives in a pineapple under the sea?

| 1. Peter Griffin

2. Scooby Doo
3. Spongebob Squarepants
4. Eric Cartman

and in an answer key, could look something like:

Who lives in a pineapple under the sea?

1. Peter Griffin
2. Scooby Doo
3. **** Spongebob Squarepants ****
4. Eric Cartman

Test

Each test has:

questions

a list of objective questions of any or all of the types previously described.

totalPoints

the sum of the points of each question

We should be able to generate a String representation of a Test as well as its answer key.

Driver

Write a program that uses your classes in order to generate both a test and an answer key, writing each to the screen. You're not expected to create an interactive, online testing program. You just need to print the test and its answer key.

Extra Credit

Send to a File (+5 points)

Add the ability to send a test and its answer key to files whose names are determined at run time.

Test Bank (+10 points)

Implement a TestBank class, which has:

- a collection of questions
- the ability to generate a Test with a given number of questions chosen randomly from the collection of questions.
- The ability to read a collection of questions from and write them to files. Reading questions from files is the most difficult part of the TestBank and it will be the source of most of the extra credit points.

What to submit

Submit all of your .java files through Canvas, and if you've completed the extra credit part of the assignment, please also provide your example input files.