Readme

Exercise: part 3

Handed by:

Itay Ostraich 316397249

Yarin Manoah 208945956

Guidelines:

- Some questions are already hard coded in the main for easier examination.
  We Don't Have equals Questions that are hard-coded.
  If you want to check equal questions you will need to add them.

- Open Questions and Multi Choice Questions with the same text are considered NOT equal.

- In case 2 (Add question and an answer), when you choose to add Multi Choice Question, after every answer you add, the system asks if you want to add another one or exit the case. It doesn't ask initially the number of answers to be added. The maximum number of answers in a multi-choice question is 8 (2 answers are added by the system: "none of the above", and "more than one answer"). In total, multi-choice questions can contain up to 10 answers, including the system's answers.

- The instruction of the program is to assume that the inputs are valid. on top of that, we programmed it to check the inputs.

- When creating multiple exam files on the same date, **the last one** is saved.

- In case we don’t have a binary file that contains a source of questions, the main program initializes the source with a static function called “initial”.
- In our Gui, when we want to clone a test, the Manger creates a clone of him and sends it over to TestGui as a clone of him.

- In addition, We add an option that gives the user a Grade after completing the test😊.
  Each question is worth the same amount of points(number of questions/100).

Part 2 :

- Initial collection is ArrayList and then the following changes

- First collection is TreeSet.
  Showing up in manager class at lines 34 & 390 (`moveQuestionsToSet(TreeSet<Question> allQuestions))`.
  The questions in the TreeSet are sorted in descending alphabetic order.
  Button `"Move Questions from ArrayList to TreeSet"`.
  Opens a window that shows the question in the TreeSet.

- Second collection is a LinkedHashSet.
  Showing up in manager class at lines 35 & 403 (`moveQuestionsToNonDuplicateSet()`).
  Duplicates are defined by the question’s text equalization.
  Button `"Move Questions from Collection to non-duplicate set"`, will only appear **after** using the button above.
  Opens a window that shows the question in the LinkedHashSet.

- Button `Move Questions from non-duplicate set to My Array List"` will only appear **after** using the button above.
  Showing up in manager class at line 410.

```
public void moveQuestionsToMyArrayList(Set<Question>
nonDuplicateQuestions)
```

Opens a window that shows the question in the MyArrayList with no duplicates.

- Button `"Print MyArrayList. Observer Button",`
  And Button “`Remove a question”`
  will only appear **after** using the button above.

  The Remove Button removes the first question from MyArrayList. As a comparison, this button also removes the first question from a regular ArrayList.
  After the removal, 2 windows of questions will pop up.

  The Print MyArrayList with Observer Button Opens a window that shows the questions in the MyArrayList with no duplicates.

- All the observer classes are in a package called “observer”.
  Contains:
  MyButton, MyLabel, MyGUI, Observer.

- If you want to compare the outputs of:
  1 – Button `Move Questions from non-duplicate set to My ArrayList"`
  2- Button `"Print MyArrayList. Observer Button"`

  First Press The 1 Button, and keep it open.

  Second Press The 2 Button, and Compere.

part 3:

- We created a new package for the Command Design Pattern,
  In the package we chose our last three buttons to execute with the command :
  `"Move Questions from ArrayList to TreeSet"`.
  `"Move Questions from Collection to non-duplicate set"`
  `"Move Questions from non-duplicate set to My Array List"`

- To see the Command functions and output, see as explained above in part 2.

- For the Memento Assignment, we chose “Question” as a primary class to work on.
  It is showing up in the Manager class on lines 89 & 97.
  (`public Memento createMemento()`
  `(public static class Memento {`

- We Created:
  A button called `"Undo Edit Question"`.
  Showing up in TestGUI class at line 198.
  (`Button undoMemento = new Button();`

To see The changes made by the Memento, you will need:

1. Press `"Edit Question"` Button to change the question text
2. Press `"Undo Edit Question"` to get back to the previous state (previous question text).
3. Each time we press the undo button we return one step backward.