EBU4201 Introductory Java Programming 2023/24

Mini Project

Task 1 [25 marks]

SuperHeroTT is a simple Graphical User Interface (GUI) application for children where they can practise their times tables (see Figure 1).

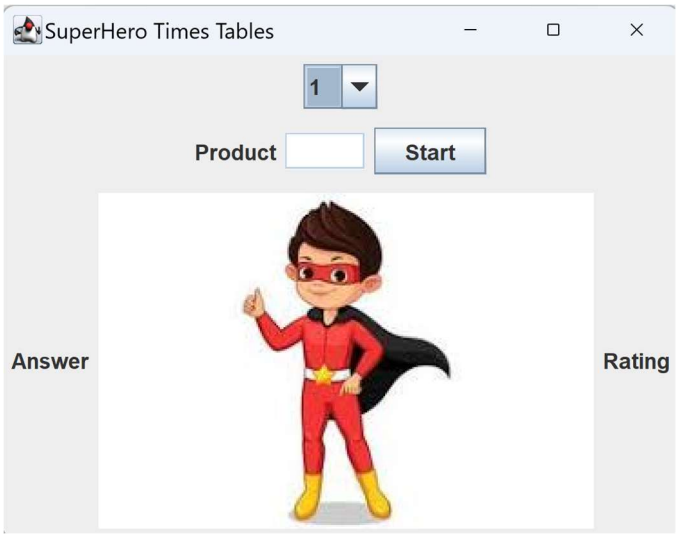

Figure 1 - First launch of SuperHeroTT

When launched, your app should look like Figure 1 - First launch of SuperHeroTT. The drop-down box at the top should consist of numbers from **1-12** (inclusive). The superhero image (**superhero.jpg**[1]) has been provided to you. The flow of the program is as follows:

1) User selects a number from the drop-down list and clicks on the **Start** button.
2) The label **Product** is replaced by a product of the form **x * n** where **x** is the selected value from the drop-down list and **n** is a random number in the range **1-12** (inclusive) AND the **Start** button text is changed to **Next**.
3) User enters their answer for the product and clicks on **Next**.
4) If the answer is correct, **Answer** is replaced by **Correct**; otherwise **Answer** is replaced by **Wrong! x * n = y** where **y** is the correct answer. See Figure 2.
5) Regardless of the answer being right or wrong, a new product should appear AND the answer box (i.e. text-field) should be cleared.
6) At the end of a round, i.e. after 5 product questions, the label **Rating** should be replaced by **You got m correct!** where **m** is the number of correct responses AND the **Next** button should change back to **Start**.
7) The above process can be repeated any number of times.

Note 1: If the user wishes, they can change their selection from the drop-down list during a round.
Note 2: If the user clicks **Next** with no text entered, there must be no run-time exceptions generated AND the product question must not change to the next one.
Note 3: The button or image must not resize when the GUI is resized.

[1] Image taken from https://www.vecteezy.com/.

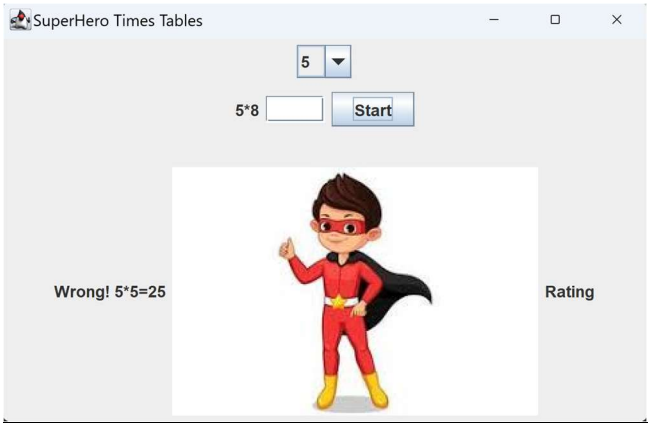

Figure 2 - Wrong answer

Hint: Consider using containers within containers and using layouts intelligently.

Note: All the necessary files should be placed in a directory called **Task1**.You can choose whether to place the image files directly under **Task1** or within a sub-directory. Whichever approach you take, the images must be displayed on the GUI without having to move the image files to different locations within your directory structure.

Also note that your application must run as expected from the command line on OpenJDK 21.0.2 without the markers having to alter any code, file locations etc. Otherwise, it will not be possible to award marks for any of the functionalities.

Task 2 [15 marks]

Your second task is to enhance the application developed in Task 1, to allow the user to select more than one multiplicand, i.e. **x** (of the product **x * n**) at the same time. The application must still function as specified in Task 1, the only difference being that **x** will be one of the selected numbers, chosen randomly, during each of the 5 questions that make up a round.

For example, say the user selected the following multiplicands: 2, 5, 6 and 12. The set of product questions can be:

5 *12
6 * 11
2 * 1
12 * 4
2 * 10

Note 1: It is possible that not all selected multiplicands will appear in a given round.
Note 2: The program must ensure that the users select at least one multiplicand. There is no maximum limit so users can select any number of multiplicands (up to 12).

Hint 1: Use another appropriate component in place of the drop-down list that was used in Task 1.

Note: All the necessary files (including any reused ones from Task 1) should be placed in a directory called **Task2**.

Documentation [10 marks]

Your submitted work must include:

a. Generated Javadocs (for all Java files)
b. Internal comments in your code (for all Java files)
c. User Manual. This should be no more than 2 pages and describe, in your own words, how to run the program (both how to start and how to use it).

Note: All documentation files should be placed in a directory called **Documentation**.

Extra Credit [5 marks]

Extra marks from this section can be used to top up your final grade for this project. The maximum mark you can achieve is still 50.

Further enhance your application in the following two ways:

a) Include a timer, i.e. at the end of a round, replace **Answer** with **You took t seconds** where **t** is the time taken to complete the round, in seconds. See an illustrative example in Figure 3.

Figure 3 - Timer

Hint: You are provided with the Java code (TimerDemo.java) for a simple timer using javax.swing.Timer. Make use of this code in your program.

Note: if you wish you can visually display a countdown on your GUI. Alternatively you can simply print out to the console, similar to TimerDemo.java.

b) Set the answer box (i.e. text-field) in focus every time the user is expected to type an answer so that they do not need to click on the answer box.

Hint: look up methods of the text-field (e.g. JTextField) class in the Java API.

Note: All the necessary files (including any reused ones from Task 1 and Task 2) should be placed in a directory called **ExtraCredit**.

Important notes:

1. This is an **individual** piece of work.

2. All three directories must be included in a zip file. The filename must be your **QM Student Number**.

3. You should design your classes properly, following object-oriented principles. E.g., do NOT write everything in the **main** method, keep code repetition to a minimum (i.e., use methods), do NOT use **static** methods unless there is a good reason. There will be marks allocated for good program design.