EBU4201: Mini-Project Coursework

Module code: **EBU4201**
Module title: **Introductory Java Programming**
Hand-out date: **5th May 2017**
Hand-in date: **29th May 2017**
Marks available: **40**
Feedback: **A marksheet will be provided with feedback comments and a mark out of 40.**

Task 1 [20 marks]

`JungleParty` is a simple application used by children to practise their counting skills; see Figure 1. When launched, a random number of animal images ranging from 1 to 10 are displayed and the user is expected to enter a number, i.e. the number of animals on display, in the text field below. When the user clicks on the button 'Check!', one of two things should happen:

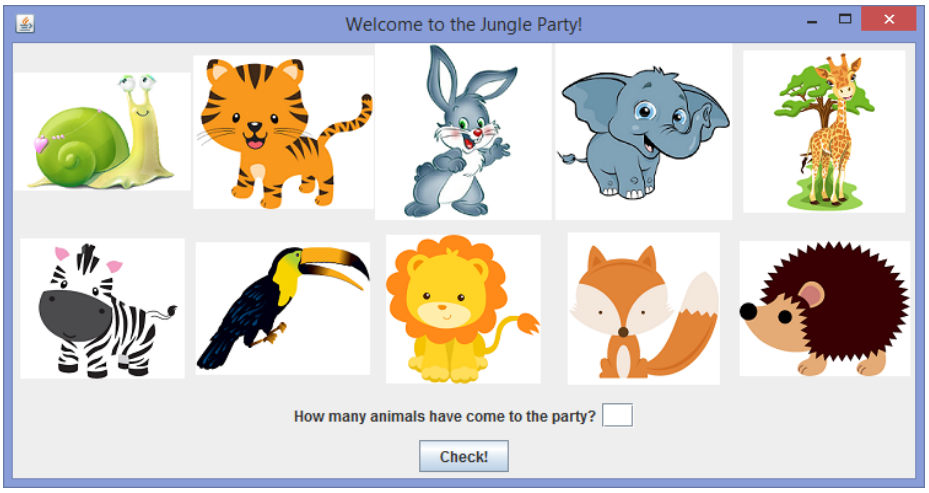

Figure 1

Case 1: The number entered is the same as the number of animal images displayed. In this case,
i. the text displayed changes to 'Correct! How many animals are in the party now?';
ii. the number of animal images displayed changes to a number between 1 and 10 (inclusive); see Figure 2 for an example.

Case 2: The number entered is NOT the same as the number of animal images displayed. In this case,
i. the text displayed changes to 'Wrong! Try again!';
ii. the number of animal images displayed does NOT change.

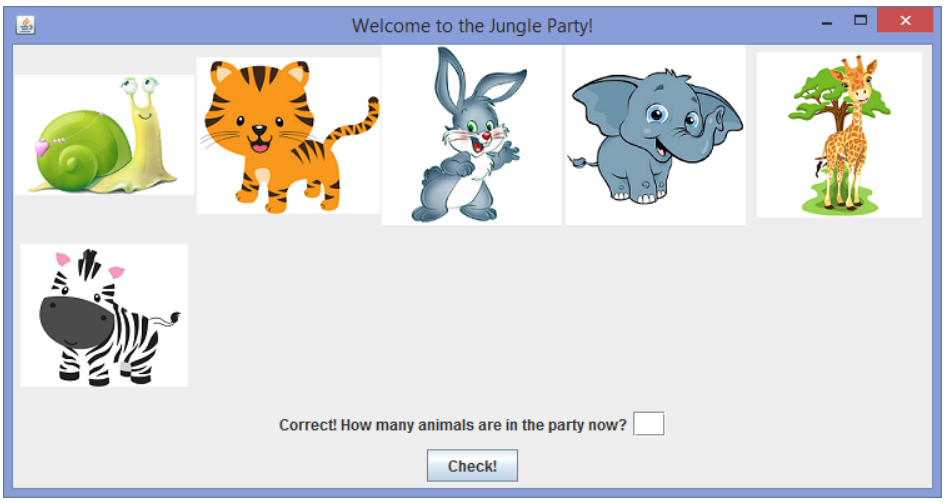

Figure 2

You are required to implement `JungleParty` as a Java application. Your application must satisfy the specific requirements given below:

a) The title of the top-level container must be 'Welcome to the Jungle Party!'.
b) The initial text displayed should be 'How many animals have come to the party?' See Figure 1.
c) The maximum number of animal images per row is FIVE. See Hint 1.
d) The text field should be wide enough to display at least TWO characters.
e) The button 'Check!' must not resize when the GUI is resized. See Hint 2.
f) When first launched, exactly 10 animal images will be displayed. However, whenever a correct answer is given, the number displayed should change to a random number between 1 and 10 (inclusive). See Hint 3 and Hint 4.
Note: It is possible for the next number to be the same as the current number.
g) Nothing should happen if the user clicks the 'Check!' button without entering anything in the text field, i.e. no errors should be thrown in this case.
Note: You can assume that only a numeric value will be entered into the text field.

Hint 1: Use an array of `JLabel` components. The following constructor may be helpful:

```
public JLabel(Icon image)
```

A set of animal images are provided in a zip file called jungle.zip. You must use these images only. The given pattern of filenames facilitates the use of loops so you are advised to retain these filenames.

Hint 2: Consider using containers within other containers and using layouts intelligently.

Hint 3: Suggested approach for displaying images: look up the following method in the class `javax.swing.JLabel`,

```
public void setIcon(Icon icon)
```

Hint 4: Suggested approaches for displaying a variable number of images: use classes `java.util.Random` OR `java.lang.Math`.

Note: All the necessary files must be placed in a directory called `Task1`. You can choose whether to place the image file 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 file to different locations within your directory structure. Also note that your application must run as expected from the command line.

Task 2 [5 marks]

You may notice that entering a non-numeric value and clicking the 'Check!' button will cause a run-time error. Therefore, your second task is to improve the application developed in Task 1 to ensure that the user is only allowed to supply valid input values, i.e. a number between 1 and 10 (inclusive). The application must still function as specified in Task 1.

Hint: Use another appropriate component in place of the text field.

Note 1: You may remove the 'Check!' button if you wish to.

Note 2: All the necessary files (including any reused ones from Task 1) must be placed in a directory called `Task2`.

Task 3 [10 marks]

You are now required to facilitate removing animals from the Jungle Party, i.e. clicking on any random animal image should make it disappear. The application must still function as specified in Task 2. In addition, the following requirements must be satisfied:

a) Add a new instruction at the bottom to say 'Click on animals you wish to kick out of the party!', in red text. Both lines of text should appear center aligned.
b) If an animal is removed from the party, change the label text to 'Animal gone! How many animals are in the party now?'.
c) The animal count input by the user should be checked against the number of animals currently on display, i.e. any removed animals need to be deducted.
d) It should be possible to click on any number of animal images until there are none left, i.e. the valid input range must now include the value zero.
e) It should NOT be possible to click on a component with no animal image (i.e. if it has already been clicked).
f) As before, once the correct number of animals has been input by the user, the number of animals displayed will change for a new round.

Hint: You will need to use a clickable type of component to display the animal images.

Note: All the necessary files (including any reused ones from Task 1 and Task 2) must be placed in a directory called `Task3`.

Documentation [5 marks]

You must include:

a. Generated Javadocs
b. Internal comments in your code.
c. User Manual. This should be no more than 2 A4 pages and include instructions on how to run the program (i.e. both how to start and how to use it).

Note: All documentation files must be placed in a directory called `Documentation`.

Extra Credit [3 marks]

Extra marks from this section can be used to top up your final grade for this project. However, the maximum mark is still 40.

Further improve your application such that the maximum number of animal images displayed can be any number between 10 and 20 (inclusive), specified as a command line argument. E.g. assuming your class is called `JunglePartyExtra`, the command

```
Java JunglePartyExtra 15
```

will launch a GUI similar to that in Figure 1 with a total of 15 animal images displayed. Whenever a correct answer is entered, the number of animal images will change to any number between 1 and 15 (inclusive). The maximum number of images per row must still be FIVE.

If no command line argument is given OR a number outside the valid range is given, the program must terminate, printing out an appropriate error message to the console. You can assume that only a numeric value will be given as the argument.

The application must still function as specified in Task 3.

Note 1: All the necessary files (including any reused ones from Task 1 - Task 3) must be placed in a directory called `ExtraCredit`.

Note 2: The 10 image files are to be reused to make up 20 animal images.

Note 3: No user action must generate any run-time errors!

Submission Instructions

1) You must zip all directories together, i.e. `Task1`, `Task2`, `Task3` and `Documentation` (also `ExtraCredit` if applicable).

2) Name your .zip file **2015xxxxxx.zip**, where **2015xxxxxx** is your QMUL student number.