University of Nebraska – Kearney
CSIT 150: Object Oriented Programming
HW4 – GUI

100 points possible

Objectives

1. Practice creating a graphic interface and responding to user events.
2. Practice dealing with strings.
3. Practice reading/writing a file.

Requirements

1. The homework is expected to be submitted on time. Late work will be accepted; however you lose 10% of your points for each late day.
2. You can get help from the instructor, the tutor, or other students. However, the homework must be finished by yourself, and you should indicate it on the paper if you got any help.
3. The first 12 questions should be answered in a document and submitted through Blackboard.
4. The programming part should be completed in Java and submitted through Blackboard.

Multiple Choice: Questions 1-15 are multiple-choice questions. Each one has one correct answer.

1. Programs that operate in a GUI environment must be _____.

   a. event driven
   b. in color
   c. dialog boxes
   d. layout managers

2. One important difference between the command line interface and a GUI interface is that _____.

   a. in command line the background color of the monitor will always be black but in a GUI it can be any color
   b. in a GUI environment the user determines the order in which things happen while the user has no control over the order or events in a command line interface
   c. users must type information in a command line interface but, when using a GUI, typing is never required
   d. All of these are important differences.
   e. None of these are an important difference. (There are important differences, but these options are not correct.)

3. To display an image in a JavaFX application you must _____.

   a. include either the `Image` or the `ImageView` class
   b. include both the `setImage` and the `ViewImage` classes
   c. include both the `Image` and the `ImageView` classes
   d. include either the `setImage` or the `ViewImage` class

4. For this question, assume `myView` references an `ImageView` object. In order to preserve the image's aspect ratio (so it does not appear stretched or distorted), you should use which of the following?

   a. `myView.setPreserveRatio(false);`
   b. `myView.setPreserveRatio(true);`
   c. `myView.setPreserveRatio();`
   d. Either (a) or (c) would work.

5. How would a stylesheet named `javafxstyles.css` be applied to a JavaFX application, assuming that `scene` is the variable that refers to a `Scene` object?

   a. `scene.getStyles().add("javafxstyles.css");`
   b. `scene.getStylesheets().add(javafxstyles.css);`
   c. `scene.getStylesheets().add("javafxstyles.css");`
   d. `scene.addStylesheets().get("javafxstyles.css");`

6. To apply specific styles to all of the nodes in a scene, use the _____ selector.

   a. `.stage` b. `.top` c. `.root` d. `.all`

7. Which of the following creates a custom style class that will allow a `Button` control to appear with a blue background and yellow text?

   a.
```
.button-color {
     -fx- background-color: blue;
     -fx- text-fill: yellow;
}
```
   b.
```
.button-color {
     -fx- bgcolor: blue;
     -fx- textfill: yellow;
}
```
   c.
```
.blue-button {
     background-color: blue;
     text-fill: yellow;
}
```
   d.
```
.blue-button {
     -fx- background: blue;
     -fx- text: yellow;
}
```

8. Adding `RadioButton` controls to a _____ object creates a mutually exclusive relationship between them.

   a. `MutualGroup`
   b. `RadioGroup`
   c. `ToggleGroup`
   d. `ExcludeGroup`

9. What happens when the following code is executed?

```
ComboBox<string> myComboBox = new ComboBox<>();
myComboBox.getItems().addAll(5, 10, 15, 20);
```

   a. A `ComboBox` displaying the numbers `5`, `10`, `15`, and `20` will be created.
   b. The values `5`, `10`, `15`, and `20` will be added to a `ComboBox` named `myComboBox`.
   c. The values `5`, `10`, `15`, and `20` will be converted to strings and added a new `ComboBox` named `myComboBox`.
   d. A compiler error will occur.

10. To build a menu system you must _____.

    a. create a `MenuBar` object
    b. create the `Menu` objects and register an event handler for each menu item object
    c. Add the `MenuBar` object to the scene graph
    d. All of these are necessary steps.

11. Which of the following creates a blue circle centered at `X = 50`, `Y = 50` with a radius of `50`?

    a.
    ```
    Circle blueCircle = new Circle(50, 50, 50);
    blueCircle.setFill(Color.BLUE);
    ```

    b.
    ```
    Circle blueCircle = new Circle(50, 50);
    blueCircle.setStroke(Color.BLUE);
    ```

    c.
    ```
    Circle blueCircle = new Circle(50, 50, 50, Color.BLUE);
    ```

    d.
    ```
    Circle blueCircle = new blueCircle(50);
    blueCircle.setColor(BLUE);
    ```

12. Any time a user presses a key, a _____ event occurs.

    a. `KEY_TYPED`
    b. `KEY_PRESSED`
    c. `KEY_RELEASED`
    d. `KEY_EVENT`

13. What does the following code snippet do?

    ```
    Circle myCircle = new Circle(50, 50, 25);
    TranslateTransition rtrans =
        new TranslateTransition(new Duration(5000), myCircle);
    ```

    a. It creates a circle with center point `(50, 25)`, radius = `50`, and duration = 5000 seconds.
    b. It creates a circle with center point `(50, 50)`, radius = `25`, and duration = 5 seconds.
    c. It creates a circle with center point `(25, 50)`, radius = `25`, and duration = 5 seconds.
    d. It creates a circle with center point `(50, 25)`, radius = `50`, and duration = 50 seconds.

14. Select all that apply. Which of the following are mouse event types?

    a. `MOUSE_CLICKED`
    b. `MOUSE_MOVED`
    c. `MOUSE_LIFTED`
    d. `MOUSE_PRESSED`

Programming Challenge

Either complete the robot maze programming challenge OR the Connect-4 programming challenge. Each of these are described in separate files. You can earn bonus points for completing both of the challenges.

15. [62 points] Correctness: Your code should be able to function correctly according to the description of the above requirements.

16. [10 points] Style and Documentation: You code should meet the style and documentation requirements listed below:
    - Use indentation to show the logical structure of your code.
    - Use blank lines to separate code blocks and give each code block a brief comment.
    - Give JavaDoc comments to each class and each method.
    - Include your name and algorithm in your program's heading documentation.
    - Output your name as part of the output.

Deliveries

- Finished program.
- Solution to questions 1-14.