Chapter 14 JavaFX Basics

Introduction to Java Programming by Y. Daniel Leung

Section 14.2 JavaFX vs Swing and AWT

14.1 Why is JavaFX preferred?

- [x] A. JavaFX is much simpler to learn and use for new Java programmers.
- [x] B. JavaFX provides a multi-touch support for touch-enabled devices such as tablets and smart phones.
- [x] C. JavaFX has a built-in 3D, animation support, video and audio playback, and runs as a standalone application or from a browser.
- [x] D. JavaFX incorporates modern GUI technologies to enable you to develop rich Internet applications.

Section 14.3 The Basic Structure of a JavaFX Program

14.2 Every JavaFX main class _____.

- [ ] A. implements javafx.application.Application
- [x] B. extends javafx.application.Application
- [x] C. overrides start(Stage s) method
- [ ] D. overrides start() method

14.3 Which of the following statements are true?

- [x] A. A primary stage is automatically created when a JavaFX main class is launched.
- [x] B. You can have multiple stages displayed in a JavaFX program.
- [x] C. A stage is displayed by invoking the show() method on the stage.
- [ ] D. A scene is placed in the stage using the addScene method
- [x] E. A scene is placed in the stage using the setScene method

14.4 What is the output of the following JavaFX program?

```
import javafx.application.Application;
import javafx.stage.Stage;

public class Test extends Application {
  public Test() {
    System.out.println("Test constructor is invoked.");
  }

  @Override // Override the start method in the Application class
  public void start(Stage primaryStage) {
```

```
    System.out.println("start method is invoked.");
  }

  public static void main(String[] args) {
    System.out.println("launch application.");
    Application.launch(args);
  }
}
```

- A. launch application. start method is invoked.
- B. start method is invoked. Test constructor is invoked.
- C. Test constructor is invoked. start method is invoked.
- D. launch application. start method is invoked. Test constructor is invoked.
- E. launch application. Test constructor is invoked. start method is invoked.

Section 14.4 Panes, UI Controls, and Shapes

14.5 Which of the following statements are true?

- A. A Scene is a Node.
- B. A Shape is a Node.
- C. A Stage is a Node.
- D. A Control is a Node.
- E. A Pane is a Node.

14.6 Which of the following statements are true?

- A. A Node can be placed in a Pane.
- B. A Node can be placed in a Scene.
- C. A Pane can be placed in a Control.
- D. A Shape can be placed in a Control.

14.7 Which of the following statements are correct?

- A. new Scene(new Button("OK"));
- B. new Scene(new Circle());
- C. new Scene(new ImageView());
- D. new Scene(new Pane());

14.8 To add a circle object into a pane, use _____.

- A. pane.add(circle);

- [ ] B. pane.addAll(circle);
- [x] C. pane.getChildren().add(circle);
- [x] D. pane.getChildren().addAll(circle);

14.9 Which of the following statements are correct?

- [x] A. Every subclass of Node has a no-arg constructor.
- [x] B. Circle is a subclass of Node.
- [x] C. Button is a subclass of Node.
- [x] D. Pane is a subclass of Node.
- [ ] E. Scene is a subclass on Node.

Section 14.5 Binding Properties

14.10 Which of the following are binding properties?

- [ ] A. Integer
- [ ] B. Double
- [x] C. IntegerProperty
- [x] D. DoubleProperty
- [ ] E. String

14.11 Which of the following can be used as a source for a binding properties?

- [ ] A. Integer
- [ ] B. Double
- [x] C. IntegerProperty
- [x] D. DoubleProperty
- [ ] E. String

Check Answer for Question 11

14.12 Suppose a JavaFX class has a binding property named weight of the type DoubleProperty. By convention, which of the following methods are defined in the class?

- [x] A. public double getWeight()
- [x] B. public void setWeight(double v)
- [x] C. public DoubleProperty weightProperty()
- [ ] D. public double weightProperty()

E. public DoubleProperty WeightProperty()

14.13 What is the output of the following code?

```
import javafx.beans.property.IntegerProperty;
import javafx.beans.property.SimpleIntegerProperty;

public class Test {
  public static void main(String[] args) {
    IntegerProperty d1 = new SimpleIntegerProperty(1);
    IntegerProperty d2 = new SimpleIntegerProperty(2);
    d1.bind(d2);
    System.out.print("d1 is " + d1.getValue()
      + " and d2 is " + d2.getValue());
    d2.setValue(3);
    System.out.println(", d1 is " + d1.getValue()
      + " and d2 is " + d2.getValue());
  }
}
```

A. d1 is 2 and d2 is 2, d1 is 3 and d2 is 3

B. d1 is 2 and d2 is 2, d1 is 2 and d2 is 3

C. d1 is 1 and d2 is 2, d1 is 1 and d2 is 3

D. d1 is 1 and d2 is 2, d1 is 3 and d2 is 3

14.14 What is the output of the following code?

```
import javafx.beans.property.IntegerProperty;
import javafx.beans.property.SimpleIntegerProperty;

public class Test {
  public static void main(String[] args) {
    IntegerProperty d1 = new SimpleIntegerProperty(1);
    IntegerProperty d2 = new SimpleIntegerProperty(2);
    d1.bindBidirectional(d2);
    System.out.print("d1 is " + d1.getValue()
      + " and d2 is " + d2.getValue());
    d1.setValue(3);
    System.out.println(", d1 is " + d1.getValue()
      + " and d2 is " + d2.getValue());
  }
}
```

A. d1 is 2 and d2 is 2, d1 is 3 and d2 is 3

B. d1 is 2 and d2 is 2, d1 is 2 and d2 is 3

C. d1 is 1 and d2 is 2, d1 is 1 and d2 is 3

D. d1 is 1 and d2 is 2, d1 is 3 and d2 is 3

Section 14.6 Common Properties and Methods for Nodes

14.15 Which of the following statements correctly sets the fill color of circle to black?

- [x] A. `circle.setFill(Color.BLACK);`
- [ ] B. `circle.setFill(Color.black);`
- [x] C. `circle.setStyle("-fx-fill: black");`
- [ ] D. `circle.setStyle("fill: black");`
- [ ] E. `circle.setStyle("-fx-fill-color: black");`

14.16 Which of the following statements correctly rotates the button 45 degrees counterclockwise?

- [ ] A. `button.setRotate(45);`
- [ ] B. `button.setRotate(Math.toRadians(45));`
- [x] C. `button.setRotate(360 - 45);`
- [x] D. `button.setRotate(-45);`

Section 14.7 The Color Class

14.17 Which of the following statements correctly creates a Color object?

- [ ] A. `new Color(3, 5, 5, 1);`
- [x] B. `new Color(0.3, 0.5, 0.5, 0.1);`
- [ ] C. `new Color(0.3, 0.5, 0.5);`
- [x] D. `Color.color(0.3, 0.5, 0.5);`
- [x] E. `Color.color(0.3, 0.5, 0.5, 0.1);`

Section 14.8 The Font Class

14.18 Which of the following statements correctly creates a Font object?

- [x] A. `new Font(34);`
- [x] B. `new Font("Times", 34);`
- [x] C. `Font.font("Times", 34);`
- [x] D. `Font.font("Times", FontWeight.NORMAL, 34);`
- [x] E. `Font.font("Times", FontWeight.NORMAL, FontPosture.ITALIC, 34);`

14.19 Which of the following statements are correct?

- [x] A. A Color object is immutable.
- [x] B. A Font object is immutable.

C. You cannot change the contents in a Color object once it is created.

D. You cannot change the contents in a Font object once it is created.

Section 14.9 The Image and ImageView Classes

14.20 Which of the following statements correctly creates an ImageView object?

A. `new ImageView("http://www.cs.armstrong.edu/liang/image/us.gif");`

B. `new ImageView(new Image("http://www.cs.armstrong.edu/liang/image/us.gif"));`

C. `new ImageView("image/us.gif");`

D. `new ImageView(new Image("image/us.gif"));`

14.21 Analyze the following code:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.geometry.Insets;
import javafx.stage.Stage;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;

public class Test extends Application {
  @Override // Override the start method in the Application class
  public void start(Stage primaryStage) {
    // Create a pane to hold the image views
    Pane pane = new HBox(10);
    pane.setPadding(new Insets(5, 5, 5, 5));
    Image image = new Image("www.cs.armstrong.edu/liang/image/us.gif");
    pane.getChildren().addAll(new ImageView(image), new ImageView(image
));

    // Create a scene and place it in the stage
    Scene scene = new Scene(pane);
    primaryStage.setTitle("ShowImage"); // Set the stage title
    primaryStage.setScene(scene); // Place the scene in the stage
    primaryStage.show(); // Display the stage
  }

  /**
   * The main method is only needed for the IDE with limited
   * JavaFX support. Not needed for running from the command line.
   */
  public static void main(String[] args) {
    launch(args);
  }
}
```

A. The program runs fine and displays two images.

B. new Image("www.cs.armstrong.edu/liang/image/us.gif") must be replaced by new Image("http://www.cs.armstrong.edu/liang/image/us.gif").

C. The image object cannot be shared by two ImageViews.

D. The addAll method needs to be replaced by the add method.

Section 14.10 Layout Panes

14.22 To add a node into a pane, use _____.

A. pane.add(node);

B. pane.addAll(node);

C. pane.getChildren().add(node);

D. pane.getChildren().addAll(node);

14.23 To add two nodes node1 and node2 into a pane, use _____.

A. pane.add(node1, node2);

B. pane.addAll(node1, node2);

C. pane.getChildren().add(node1, node2);

D. pane.getChildren().addAll(node1, node2);

14.24 To remove a node from the pane, use _____.

A. pane.remove(node);

B. pane.removeAll(node);

C. pane.getChildren().remove(node);

D. pane.getChildren().removeAll(node);

14.25 To remove two nodes node1 and node2 from a pane, use _____.

A. pane.remove(node1, node2);

B. pane.removeAll(node1, node2);

C. pane.getChildren().remove(node1, node2);

D. pane.getChildren().removeAll(node1, node2);

14.26 Which of the following statements are correct to create a FlowPane?

A. new FlowPane()

B. new FlowPane(4, 5)

C. new FlowPane(Orientation.VERTICAL);

D. new FlowPane(4, 5, Orientation.VERTICAL);

14.27 To add a node to the the first row and second column in a GridPane pane, use _____.

- A. pane.getChildren().add(node, 1, 2);
- B. pane.add(node, 1, 2);
- C. pane.getChildren().add(node, 0, 1);
- D. pane.add(node, 0, 1);
- E. pane.add(node, 1, 0);

14.28 To add two nodes node1 and node2 to the the first row in a GridPane pane, use _____.

- [x] A. pane.add(node1, 0, 0); pane.add(node2, 1, 0);
- [ ] B. pane.add(node1, node2, 0);
- [x] C. pane.addRow(0, node1, node2);
- [ ] D. pane.addRow(1, node1, node2);
- [ ] E. pane.add(node1, 0, 1); pane.add(node2, 1, 1);

14.29 To place a node in the left of a BorderPane p, use _____.

- A. p.setEast(node);
- B. p.placeLeft(node);
- C. p.setLeft(node);
- D. p.left(node);

14.30 To place two nodes node1 and node2 in a HBox p, use _____.

- A. p.add(node1, node2);
- B. p.addAll(node1, node2);
- C. p.getChildren().add(node1, node2);
- D. p.getChildren().addAll(node1, node2);

14.31 Analyze the following code:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.scene.layout.HBox;
import javafx.scene.shape.Circle;

public class Test extends Application {
  @Override // Override the start method in the Application class
  public void start(Stage primaryStage) {
```

```
    HBox pane = new HBox(5);
    Circle circle = new Circle(50, 200, 200);
    pane.getChildren().addAll(circle);

    circle.setCenterX(100);
    circle.setCenterY(100);
    circle.setRadius(50);
    pane.getChildren().addAll(circle);

    // Create a scene and place it in the stage
    Scene scene = new Scene(pane);
    primaryStage.setTitle("Test"); // Set the stage title
    primaryStage.setScene(scene); // Place the scene in the stage
    primaryStage.show(); // Display the stage
  }

  /**
   * The main method is only needed for the IDE with limited
   * JavaFX support. Not needed for running from the command line.
   */
  public static void main(String[] args) {
    launch(args);
  }
}
```

- (●) A. The program has a compile error since the circle is added to a pane twice.
- ( ) B. The program has a runtime error since the circle is added to a pane twice.
- ( ) C. The program runs fine and displays one circle.
- ( ) D. The program runs fine and displays two circles.

Section 14.11 Shapes

14.32 The _____ properties are defined in the javafx.scene.shape.Shape class.

- [x] A. stroke
- [x] B. strokeWidth
- [x] C. fill
- [ ] D. centerX

14.33 The _____ properties are defined in the javafx.scene.text.Text class.

- [x] A. text
- [x] B. x
- [x] C. y
- [x] D. underline

- [x] E. strikethrough

14.34 The _____ properties are defined in the javafx.scene.shape.Line class.

- [x] A. x1
- [x] B. x2
- [x] C. y1
- [x] D. y2
- [ ] E. strikethrough

14.35 The _____ properties are defined in the javafx.scene.shape.Rectangle class.

- [x] A. width
- [x] B. x
- [x] C. y
- [x] D. height
- [x] E. arcWidth

14.36 The _____ properties are defined in the javafx.scene.shape.Ellipse class.

- [x] A. centerX
- [x] B. centerY
- [x] C. radiusX
- [x] D. radiusY

14.37 To construct a Polygon with three points x1, y1, x2, y2, x3, and y3, use _____.

- [x] A. new Polygon(x1, y1, x2, y2, x3, y3)
- [ ] B. new Polygon(x1, y2, x3, y1, y2, y3)
- [x] C. Polygon polygon = new Polygon(); polygon.getPoints().addAll(x1, y1, x2, y2, x3, y3)
- [ ] D. Polygon polygon = new Polygon(); polygon.getPoints().addAll(x1, y2, x3, y1, y2, y3)

14.38 To construct a Polyline with three points x1, y1, x2, y2, x3, and y3, use _____.

- [x] A. new Polyline(x1, y1, x2, y2, x3, y3)

- [ ] B. `new Polyline(x1, y2, x3, y1, y2, y3)`
- [x] C. `Polyline polyline = new Polygon(); polyline.getPoints().addAll(x1, y1, x2, y2, x3, y3)`
- [ ] D. `Polyline polyline = new Polygon(); polyline.getPoints().addAll(x1, y2, x3, y1, y2, y3)`

14.39 Assume p is a Polygon, to add a point (4, 5) into p, use _____.

- [ ] A. `p.getPoints().add(4); p.getPoints().add(5);`
- [x] B. `p.getPoints().add(4.0); p.getPoints().add(5.0);`
- [ ] C. `p.getPoints().addAll(4, 5);`
- [x] D. `p.getPoints().addAll(4.0, 5.0);`