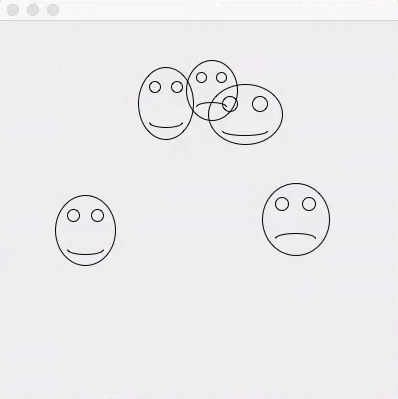
Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Score \_\_\_\_\_\_\_ / 28

CS 245: Object-Oriented Programming

Homework 3

1. Which of the following is a difference between ArrayList objects and arrays?
   1. an array can store only one type of object, but an array list can store a variety of kinds of objects.
   2. an array is fixed in size, whereas an array list will automatically resize based on how many elements are stored in it.
   3. an array cannot be used to support polymorphism, but an array list can.
   4. you don’t have to create the array itself, whereas you do have to create the array list object
2. What is not true about primitive data types in Java?
   1. we use primitive type wrappers to help convert among the various types and to give the primitive data types functionality.
   2. primitive data types hold both the data and methods for working with that data.
   3. they are the only non-object data types in Java.
   4. The Scanner class has functions for reading primitive data type values.
3. What is the job of a layout manager?
   1. To render user interface components in a container.
   2. To render a drawing on a canvas.
   3. To position user interface components in a container.
   4. To position a frame on a desktop.
4. The layout manager that positions controls based on north, south, east, west, and center sections is called
   1. BorderLayout
   2. GridLayout
   3. FlowLayout
   4. NullLayout
5. Which of the following is a difference between lightweight and heavyweight components.
   1. Lightweight components render themselves using paint, but heavyweight components render themselves using paintComponent.
   2. Lightweight components render themselves using paintComponent, but heavyweight components render themselves using paint.
   3. Lightweight components can’t contain other components, but heavyweight components can.
   4. Lightweight components can’t have layout managers, but heavyweight components can.
6. What is the purpose of calling super.paint(g) at the beginning of a JFrame’s paint function?
   1. It makes sure all the drawing tasks, like g.drawOval, are performed.
   2. It makes sure that the Graphics object is created so that we can draw on it.
   3. It makes sure that the background and borders and outlines of graphical user interface components are drawn.
   4. It reduces memory leaks associated with drawing shapes on the screen.
7. Which of the following layout managers would be best for arranging the keys of a calculator?
   1. BorderLayout
   2. GridLayout
   3. FlowLayout
   4. NullLayout
8. With a Model-View-Controller arrangement, which of the following is usually true?
   1. The Controller has a data member that is an instance of the Model class as well as a data member that is an instance of the View class.
   2. The Model has a data member that is an instance of the View
   3. The Model has a data member that is an instance of the Controller
   4. The View has a data member that is an instance of the Controller
9. Which of the following represents the correct sequence of steps for how a JFrame is repainted?
   1. repaint triggers a call to paintComponent, which tells all the components owned by the frame to paint themselves.
   2. paint triggers a call to repaint, which tells all the components owned by the frame to call their paintComponent function
   3. repaint triggers a call to the paintComponent function of all of the components owned by the frame. The last component tells the frame to call its paint function
   4. repaint tells the frame to call its paint function, which, in turn, calls the paintComponent function of each of the components owned by the frame.
10. Write the command needed to position a frame of width 200 and height 300 at 100 pixels from the left and 100 pixels from the top of the screen.
11. FaceDraw Assignment (18 points)

****

For this assignment, you will create an application in Java that draws random faces on a window. The application will draw anywhere from 3 to 10 random faces each time it is run. The faces will have a random width and height, with those widths and heights set to reasonable ranges so that the faces have dimensions typical of a face. Each face is to have two eyes and a mouth. The mouth may be smiling, frowning, or in-between, and that is chosen randomly.

Here is how your program will be graded:

|  |  |
| --- | --- |
| Requirement | Points |
| The Face class is the model class. It has data members that store the width, height, x, y, and smile status. It also has constructors, get and set functions, and a toString function. | 3 |
| Your main function creates and randomly populates a list of Face objects. The Face objects are created with random width, top, left, height, and smile status. | 3 |
| You have a view class that descends from JFrame called FaceFrame. It has one component within it: a JPanel descendant object of type FacePanel that will be used to hold the FaceObjects. | 3 |
| Your main function creates the FaceFrame object, passing to it the list of Face objects. The FaceFrame, in turn, creates its FacePanel, passing to FacePanel’s constructor the list of Face objects so that it has access to them to draw them. | 3 |
| The FacePanel’s paintComponent function actually draws the Face objects where they are supposed to be and how they are supposed to look. | 3 |
| In addition to drawing the Face objects, your program calls each Face object's toString function to have the Face describe itself to the console window. This should be done from the main function. | 1 |
| Your code is sufficiently commented | 1 |
| You submit your .java file as a file named Face\_LastName.java, where you replace "LastName" with your last name. | 1 |

If your program does not work, the most you can get is half credit. In other words, if your program crashes, you will get a maximum of 9 points out of 18.

Do not copy another student’s work. I will use MOSS to detect plagiarism and will not ask for clarification if MOSS concludes you have copied another student’s work.

Tackle this problem gradually. I recommend creating the Face class first. Then, write a main function that populates a list of Face objects with random faces. Then, create the window by extending JFrame, resulting in the FaceFrame class. Then, flesh out the FacePanel class. That last part will require calculating where to put the eyes and mouth and how to draw the mouth as smiling, frowning, or in-between.

Definitely pace yourself. Do not attempt to do this in one night.

Good luck – and have fun. This is, indeed, supposed to be fun.