Lab 7: Methods
20 pts

Distribute on November 6, 2023
Due before November 12, 2023 (Sunday) at 12:00 midnight

Learning Outcomes ((CLO) vs (SO) Mapping)

- Utilize Java syntax in fundamental programming algorithms (3)vs(1)
- Recognize and apply the various control structures (5)vs(1)
- Design and implement elementary multiclass solutions (6)vs(2, 6)

Requirements

The focus of this lab is to gain experience with Java methods.

Preliminaries

1. Create an Eclipse Java project. The name of the project must be **lab07_<your FirstNameLastName>**. For example, my project would be named, lab07_PeterNg.
2. Add two Java classes named **ESPGame** and **Applications** to the package **colors** in your project. (When you enter first enter your Java class, change whatever is in your package to **colors**.)
3. Add the following comment block to the beginning of each of your classes:

```
/*
 * <your name>
 * CS 16000-01 02/03, Fall Semester 2023
 *  (Note: write either 02 or 03, depending on which is your section.)
 * Lab 7
 *
 */
```

4. Make additional documentation about the purpose of the given classes.

5. **Ask the TA to verify that you working in the lab**

Exercises
In this lab you are going to create a program to solve an upgraded version of Programming Challenges "ESP Game", Problem 18, p316, in 7th Edition, in Chapter 5 Methods of your textbook. It states:
"Write a program that tests your ESP (extrasensory perception). The program should randomly select the name of a color from the following list of words:
Blue, Yellow, Red, Green, Orange, Cyan, darkGray, lightGray,
Gray, Pink, Magenta, White, Black, Brown, Beige

To select a word, the program can generate a random number. For example, if the number is 0, the selected word is Blue; if the number is 1, the selected word is Yellow, and so forth.

Next, the program should ask the user to enter the color the computer selects. After the user has entered his or her guess, the program should display the name of the randomly selected color. The program should repeat this 10 times and then display the number of times the user correctly guessed the selected color. Modularize the program into methods that perform each major task."

1. Class **`ESPGame`** contains one private data field: declare a String variable named **chosenColor**. (That is, declare the variable right in the class, not in any of the methods.)
2. The class contains three user-defined methods named chooseColor( ), showColor( ) and guessColor( ). All the methods are **public and none of them is static.** See a more detailed description of the methods below.
3. The methods make use of the Java library class **`Color`**. Import java.awt.Color and javax.swing.* to your project. Note that certain colors in Java are represented as named constants of the Color class, for instance, Color.BLUE etc.
4. Method descriptions:

a. `chooseColor( )`

- return type: Color
- takes an int type parameter

This method returns one of the colors Color.BLUE, Color.YELLOW, Color.RED, Color.GREEN, Color.ORANGE, Color.CYAN, Color.darkGray, Color.GRAY, Color.PINK, Color.MAGENTA, Color.WHITE, Color.lightGray, Color.BLACK, according to the input parameter values 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13.

The method declares the local variables **`color`** of type **`Color`**, **`input`** of type **`int`**, and runs a switch selection. The switch is controlled by input. A typical case in the switch is like

```
case 4:
        color = Color.GREEN;
        chosenColor = "green";
        break;
…
case 7:
        color = Color.darkGray;
        chosenColor = "dark gray";
        break;
```

Note that the data field chosenColor stores the name of the chosen color in String format. The color choice for the default case in the switch is Color.BLACK.

b. `showColor()`

- void
- takes a Color type parameter

The purpose of this method is to create a small window for displaying the parameter color. Give the name **`color`** to the parameter and copy the following code into the method's body:

```
JFrame frame = new JFrame("Guess this color");
frame.setSize(200,200);
frame.setLocation(300,300);
JPanel panel = new JPanel();
panel.setBackground(color);
frame.add(panel);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

You may try 450 and 350 or other numeric values for the size and location, respectively.
Note that you do not have to study either to understand the above code fragment. However, you have to re-write the double quote operator into your program by hand, the copied character is wrong for Java.

`C.guessColor()`

- void,
- takes no parameter

This method runs a while loop to offer repeatedly the ESP game to the user. The offer is displayed on JOptionPane confirm dialog (Figure 1).

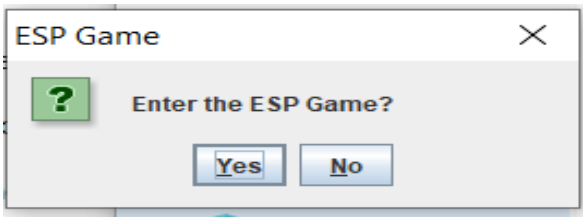

Figure 1

- The syntax to open and use a confirm dialog window has a detailed description in Lab 5. The if-else statement controlled by the window works as follows:

  For a 'No' answer, the program exits. In particular, the while loop is stopped here with a break, thus you can run the loop with the boolean literal **`true`**.

- Before the program exits, the following message in the JOptionPane dialog box is displayed (as shown in Figure 1a). And the console displays the following message:

```
No game was played.
```

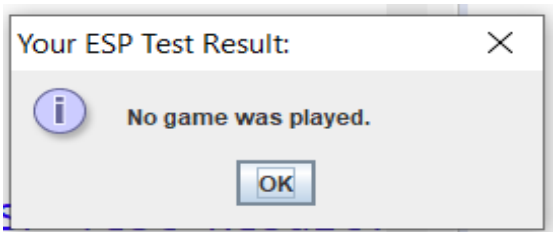

Figure 1a

For the 'Yes' answer, the method

- randomly selects an integer between 1 and 13,
- calls the chooseColor( ) method with the random number as a parameter,
- calls the showColor( ) method with the color value returned by chooseColor( ), as parameter.

As a result, the two windows of Figures 2 and 3 simultaneously appear on the screen:

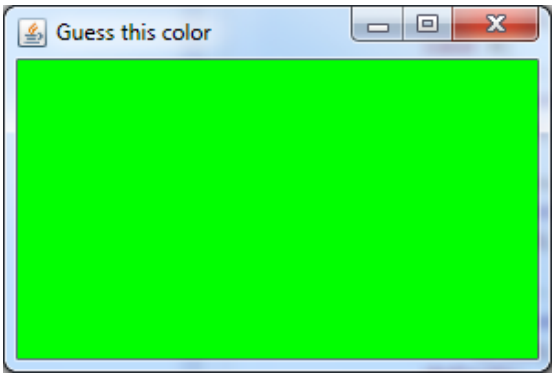

Figure 2

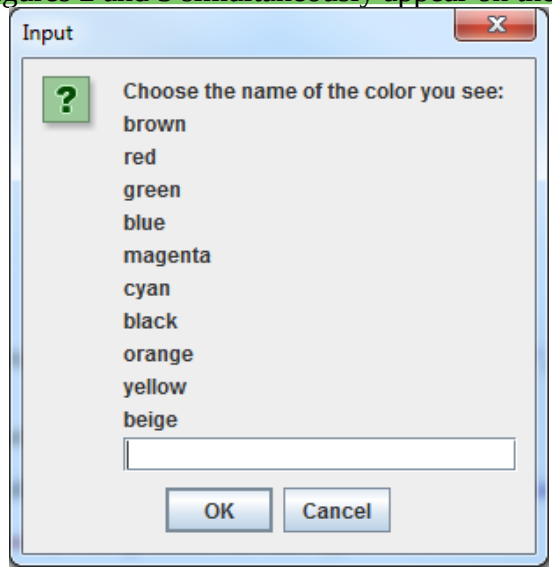

Figure 3

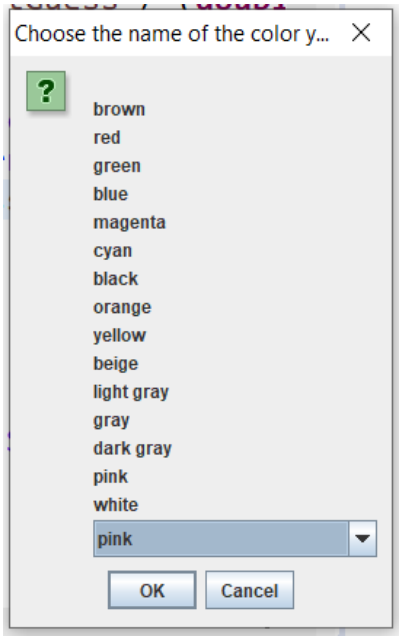

Figure 3a

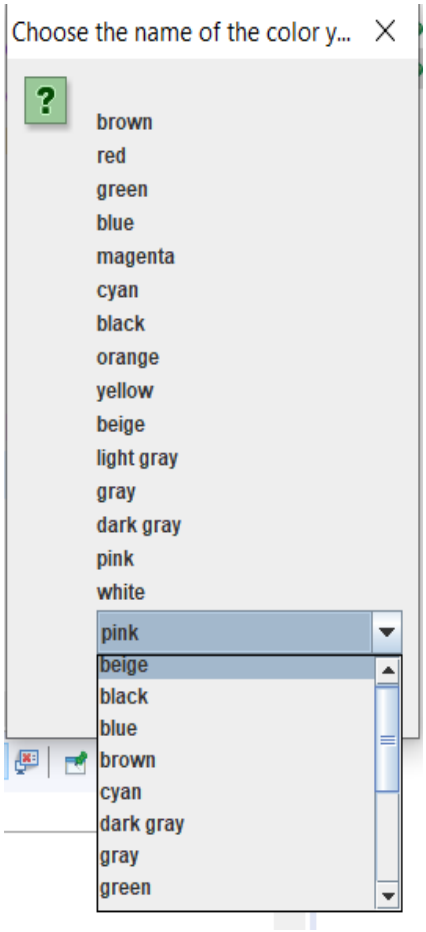

Figure 3b

Instead, you need to try to implement the following: Figure 3a allows the user to click a color from the give list in a dropdown window. instead of typing the color in the window as shown in Figure 3a. Then click on a little triangle, and Figure 3b is depicted for guessing the displayed color. This is far better than Figure 3.
The following codes could be:

```
String str01 = "Choose the name of the color you see:";

String str02 = "\nbrown\nred\ngreen\nblue\nmagenta\ncyan\nblack"
+ "\norange\nyellow\nbeige\nlight gray\ngray\ndark gray\npink"
+ "\nwhite";

String[] str03 = {"beige", "black", "blue", "brown", "cyan",
"dark gray", "gray", "green", "light gray", "magenta", "orange",
"pink", "red", "white", "yellow" };

String user_input = (String) JOptionPane.showInputDialog(null,
str02, str01, JOptionPane.QUESTION_MESSAGE, null, str03,
str03[11]);
```

Note that str03[11] is to select "pink" from the str03 to appear in the dropdown window. You need to select "green" instead of "pink".
Such a dropdown window is widely used in many applications.

- Having the user entered a choice, the method compares the selection to the chosenColor value and displays the result according to the templates of Figures 4 and 5.
- Count the number of tries (i.e., yes for Enter the ESP game? in Figure 1), the number of guessed correct (Figure 4), number of guessed wrong (Figure 5). Compute the percentage of the correct guesses. Display the results on the console as shown below, such as:

```
Out of 11 games, you have guessed 10 correctly.
The percentage for your correct guesses is 90.91.

Or

Out of 3 games, you have guessed 0 correctly.
The percentage for your correct guesses is 0.00.
```

Also, send and display messages to the JOptionPane dialog boxes as shown in Figure 6 and Figure 6a.

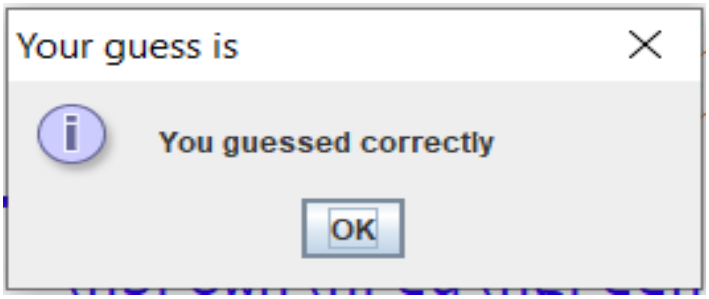

Figure 4

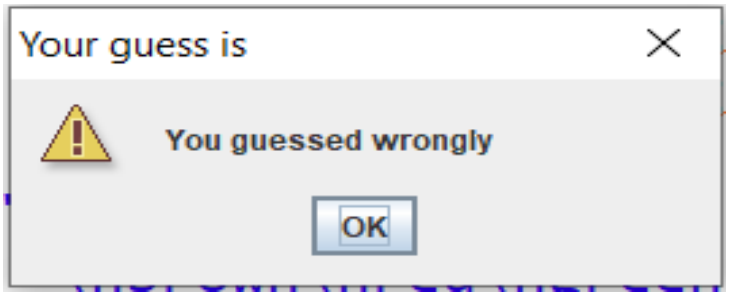

Figure 5

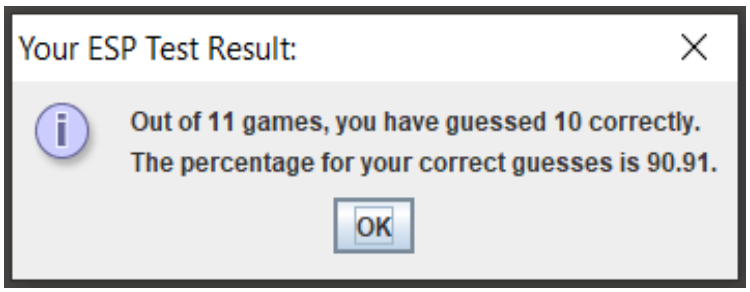

Figure 6

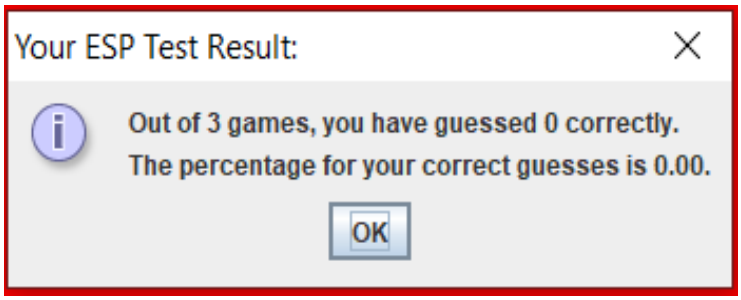

Figure 6a

5. Class **`Applications`** contains nothing but the main method for which the code is supplied below.

```
public static void main(String[] args) {
        ESPGame game = new ESPGame();
    game.guessColor();
}
```

Testing

Test your methods one by one with temporary non-random input values. Make temporary changes in the main method to call other methods of the ESPGame class, and you may add temporary printing statements to the methods tested.

Your program must follow the descriptions accurately.

Submit

- Zip your project folder and submit your assignment on Brightespace.