

```

//lab07
package lab07_PNg_ESPGame_OnColors;

public class Applications_OnColors {

    public static void main(String[] args) {
        //PROBLEM 5: Class Applications contains nothing but ...
        //Create a new ESPGame object
        ESPGame_OnColors game = new ESPGame_OnColors();

        //Start the color guessing
        game.guessColor();

    } //end of main
} //end of class

```

```

package lab07_PNg_ESPGame_OnColors; //in creating Application class,
allow to change
//package to colors. Otherwise the package will
//kept the project name lab02_PNg_ESPGame_OnColors

import java.util.Random;
//make use of the Java library class Color.
import java.awt.Color;
/*
 * AWT stands for Abstract Window Toolkit
 * It is a graphical user interface (GUI) framework that
 * provides classes for building graphical components, such

```

```

* as windows, buttons, menus and colors.
* Example classes: Color, Button, Frame, Panel, Graphics, etc.
*/

/*
import javax.swing.JFrame; // a top-level main application window.
import javax.swing.JOptionPane;
import javax.swing.JPanel; //a container for grouping components.
import javax.swing.JButton; //a push button
*/
import javax.swing.*;
/*
* Swing is a GUI toolkit in Java, built on top of AWT.
* It provides more powerful, flexible, and platform-independent
* components than AWT.
* The java.swing.* import gives you access to all Swing classes.
* such as JFrame, JPanel, JButton, JOptionPane.
*/
/**
* Peter Ng
* CS 160 - 01/02/03, Fall 2025
* Lab 7
* @author apeng
*
*/

public class ESPGame_OnColors {

    //Problem 1: Class ESPGame_Colors contain one private data field:
    //          declare a String variable named chosenColor.
    private String chosenColor;

    /**
     *
     * @param input
     * @return
     */

    //Problem 2: The class ESPGame_Colors contains three user defined
    //          methods: chooseColor(), showColor() and guessColor()

    //Problem a. chooseColor() has return type: Color and
    //          take an int type parameter

    public Color chooseColor(int input) {

```

```

//Problem a: declares the local variable color of type
Color,
//          input of type int and runs a switch selection.
//          The switch is controlled by input.

//Store the color found by the logic
Color color;

//Depending on user_input color chosen. If none of these
default to black
switch (input) {

//Problem a: A typical case in the switch is like case 4.
//          the data field chosenColor stores the name of
the
//          chosen color in String format, such as
//          chosenColor = "blue"
//Problem a: The color choice for the default case in the
switch
//          is Color.BLACK.
    case 1:
//Problem 3: certain colors in Java are
represented as named
//          constants of the Color class, e.g,
Color.BLUE etc.
        color = Color.BLUE;
        chosenColor = "blue";
        break;
    case 2:
        color = Color.YELLOW;
        chosenColor = "yellow";
        break;
    case 3:
        color = Color.RED;
        chosenColor = "red";
        break;
    case 4:
        color = Color.GREEN;
        chosenColor = "green";
        break;
    case 5:
        color = Color.ORANGE;
        chosenColor = "orange";
        break;
    case 6:

```

```

        color = Color.CYAN;
        chosenColor = "cyan";
        break;
//The following colors are not included in the assignment,except color
black.

```

```

        case 7:
            color = Color.DARK_GRAY;// Color.darkGray;
            chosenColor = "dark gray";
            break;
        case 8:
            color = Color.GRAY;
            chosenColor = "gray";
            break;
        case 9:
            color = Color.PINK;
            chosenColor = "pink";
            break;
        case 10:
            color = Color.MAGENTA;
            chosenColor = "magenta";
            break;
        case 11:
            color = Color.WHITE;
            chosenColor = "white";
            break;
        case 12:
            color = Color.LightGray;
            chosenColor = "light gray";
            break;

        default:
            color = Color.BLACK;
            chosenColor = "black";
            break;
    }//end of switch (input)

    //return the found color
    return color;

} //end of the method chooseColor(int input)

/**
 * Problem b: 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 body:
*
* @param color
*/

//Problem b. showColor() method has a void return type and
//                                     takes a Color type parameter
public void showColor(Color color) {

    //set up the JFrame to show the color to the user
    JFrame frame = new JFrame("Guess this color");

    frame.setSize(450,450);
    frame.setLocation(350,350);

    JPanel panel = new JPanel();
    panel.setBackground(color);

    frame.add(panel);
    frame.setVisible(true);

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    /* Attempt to close the color frame but fail!
       System.out.println(color);
       if (color == null)
       {
           frame = new JFrame("Guess this color");

           frame.setVisible(false);
           frame.dispose();

       }

    */

}

/**
 * Problem c: The method guessColor() has void return type and
 *             no parameter.
 *             This method runs a while loop to offer repeatedly

```

```

        *           the ESP game to the user.
        */

    public void guessColor() {

        //Problem c: 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).

        //Problem c: For 'No' answer the program exits. In
particular, the while
        //           loop is stopped with a break, thus you can
run the loop
        //           with the boolean literal true.
        //Instead use yesNo == JOptionPane.YES_OPTION to control
while-loop

        int totalGamePlayed = 0;    //Total ESP Game is played.
        int totalCorrectGame = 0;   //number of ESP games is
correctly responded.

        //Prompt an user whether entering the ESP Game.
        int yesNo = JOptionPane.showConfirmDialog(null, "Enter the
ESP Game?", "ESP Game",
                                           JOptionPane.YES_NO_OPTION);

        while (yesNo == JOptionPane.YES_OPTION)
        {

            //Problem c: For 'Yes' answer, the method
            //           randomly selects an integer
between 1 and 6.
            //           calls the chooseColor() method with
the random number
            //           as parameter,
            //           calls the showColor() method with
the color value returned
            //           by chooseColor(), as
parameter
            //           Having the user entered a choice,
the method compares the
            //           selection to the chosenColor
value and displays the

```

```

// result according to the
templates of Figure 4 and 5.

//create a Random class object.
Random rand = new Random();

//Problem c: randomly selects an integer between
1 and 13 instead of 6

//Get a random number, 1 to 13, for the color
int num = rand.nextInt(13) + 1; //range is [1,
14). i.e, [1, 13]

//Problem c: calls the chooseColor() method with
the random number
// as parameter
// calls the showColor() method with
the color value returned
// by chooseColor(), as parameter

//Show the user the random color in a small
window with title "Guess this color"
// (as shown in Figure 2)
showColor(chooseColor(num));

//prompt the user to guess the color as shown in
Figure 3.
/*
String user_input =
JOptionPane.showInputDialog("Choose the name of the color you see:" +
"\nbrown\nred\ngreen\nblue\nmagenta\ncyan\nblack\norange\nyellow\n
beige" +
"\nlight gray\ngray\ndark
gray\npink\nwhite");
*/
String str01 = "Choose the name of the color you
see: "; //for the title
String str02 =
"\nbrown\nred\ngreen\nblue\nmagenta\ncyan\nblack\norange\nyellow\n
beige" +
"\nlight gray\ngray\ndark
gray\npink\nwhite"; //for the task
//Allow user to click a color from the give list
instead of typing the color.

```

```

        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[14]);

//          System.out.println(user_input); //this statement
can shown if click the button Cancel,                                     //the
user_input has a value null.

        //Problem c: Having the user entered a choice,
the method compares the
        //          selection to the chosenColor
value and displays the
        //          result according to the
templates of Figure 4 and 5.

        //Test whether the user click on the button
Cancel. If not, then continue the process.
        //This avoids an incorrect execution of the
program.

        if (user_input != null)
        {

                //Check user whether name the right color
for the shown color.

                if(user_input.equals(chosenColor)) {
                        //Inform user if the user correctly
names the color as shown in Figure 4.
                        JOptionPane.showMessageDialog(null,
"You guessed correct", "Your guess is",
JOptionPane.INFORMATION_MESSAGE);

                                ++totalGamePlayed;
                        //increase total ESP Game played by one.
                                ++totalCorrectGame;
                        //increase number of correct ESP games played.

                }
                else {

```

```

//Inform user if the user
incorrectly names the color as shown in Figure 5.

JOptionPane.showMessageDialog(null, "You guessed wrong", "Your
guess is",

JOptionPane.WARNING_MESSAGE);

++totalGamePlayed;
//increase total ESP Game played by one.
} //end of
if(user_input.equals(chosenColor))

} //end of if part of if (user_input != null)

//Problem c: For 'No" answer, the program exits.
else
{
    break; //This is for you Cancel if
asking Guess the color.
} //end of else part of if (user_input != null)

//Prompt user whether going for the next ESP
Game.
yesNo = JOptionPane.showConfirmDialog(null,
"Enter the ESP Game?", "ESP Game",
JOptionPane.YES_NO_OPTION);

/*
//attempt to dispose the existing frame.
if (yesNo == JOptionPane.NO_OPTION)
showColor(null);
*/

} //end of while (yesNo == JOptionPane.YES_OPTION)

//The user is done playing
//Display the results
displayResult(totalGamePlayed, totalCorrectGame);

System.exit(0); //This is the way to shut down the frame of
the color.

//but this is improper approach!

} //end of guessColor() method

```

```

/**
 *
 * @param gamesPlayed
 * @param correctGuess
 */

public void displayResult(int gamesPlayed, int correctGuess){

    String str04;
    double correctGuessPercent =
100*(double)correctGuess/((double)gamesPlayed;
    if (gamesPlayed > 0)
    {
        str04 = String.format("Out of %d games, you have
guessed %d correct.\n" +
                                "The percentage for
your correct guesses is %.2f.",
                                gamesPlayed,
correctGuess, correctGuessPercent);
        System.out.println(str04);
    }
    else
    {
        System.out.println("No game was played.");
        str04 = "No game was played.";
    }
    JOptionPane.showMessageDialog(null, str04, "Your ESP Test
Result: ",
JOptionPane.INFORMATION_MESSAGE);

    } //End of displayResult() method

} //end of class ESPGame_OnColors

```