Java Swing – An introduction to User Interface

# Colour

Colours are made of red, green, and blue components, each represented by an int value that describes its intensity, ranging from \_\_\_\_\_(darkest) to \_\_\_\_\_ (lightest)

You can create a colour using the constructor:

Color (int r, int g, int b) -> r g b refers to red, green, blue respectively

**Take a look at DemoColour.java.**

Firstly, if you are using IntelliJ, the computer gives a hint as to what the colour is on the right.

Secondly, notice how Color.RED is used. Seemingly, 2 conventions are broken: a) variables are lowercase with every other word uppercase, and b) variables are supposed to be private.

This, however, is what we call a \_\_\_\_\_\_\_\_\_\_\_ – a value that is used for programming that will not change. Constants are supposed to be constant, they should not be changed. They are declared using the word final in front of the variable declaration.

All components have the ability to setBackground() and setForeGround().

## Practise

1. Take the code from the DemoPanels class, and change the colour of the start and stop buttons.
2. Change the calculator so that it is personalized, and is effectively coloured (ie. C, CE are a different colour, etc)

# Font

Similarily, font can be changed.

**Take a look at DemoFont.java.**

The constructor for Font is:

public Font(String name, int style, int size);

You can choose a font name from SansSerif, Serif, Monospaced, Dialog, and DialogInput, choose a style from Font.PLAIN (0), Font.BOLD (1), Font.ITALIC (2), and Font.BOLD+Font.ITALIC (3), and specify a font size of any positive integer.

**Take a look at FontFinder.java.**

You can find out what fonts there are in your computer using FontFinder. However, remember that not all computers support all the same fonts.

And similarly, all components have the ability to setFont().

## Practise

1. Take the code from the DemoPanels class, and change the font of the microwave time and the numbers.
2. Change the calculator so that it is personalized, and that the font is professional-looking.

# Image Icon

Components can also have icons: small, fix-sized picture as decoration. The picture MUST be GIF, JPEG, or PNG.

**Take a look at DemoImageIcon.java.**

Make sure the image can be found. If it is not found, no image shows. Thankfully, we don’t have to worry too much about FileIO exceptions.

## Practise

1. Display a frame that contains three labels. Each label displays a card (you will need to find the card images yourself).

Miscellaneous Common Properties

**Take a look at DemoCommonSwing.java.**

There are other properties, if you want to get nitpicky. Borders, cursor shapes, and hover text (Tool tip text).

These are more for the finishing touches than actual necessary parts of programming.

See Textbook page 462 – 465 if you want to know more about this.

# Scrollable Menu

Perhaps your screen cannot fit as many things are you like. You may use the JScrollPane, which allows for scrolling. It is a subclass of JPanel, also a Component.

**Take a look at DemoScrollable.java.**

# Menu

Finally, menus. This is a staple for most programs – having the ability to change settings or access qualities of the program itself.

Components we can consider:

* JMenuBar
* JMenu
* JMenuItem
  + JCheckBoxMenuItem
  + JRadioButtonMenuItem
* ButtonGroup

**Take a look at DemoJMenu.java.**

We’ll get more into the interaction of the user and the program later on.

## Practise

1. Take the previous practise question, and add a menu with File and About menus. In the file menu, have options for new game, save, and quit. In the about menu, have options for help, game rules, and high scores.

# Work

Please work on Programming Exercises from the textbook (pg 476 -478), numbers 9, 10, 11, and 13.