Assignment List vs Set with GUI

CSIS-1410

Learning Objectives :

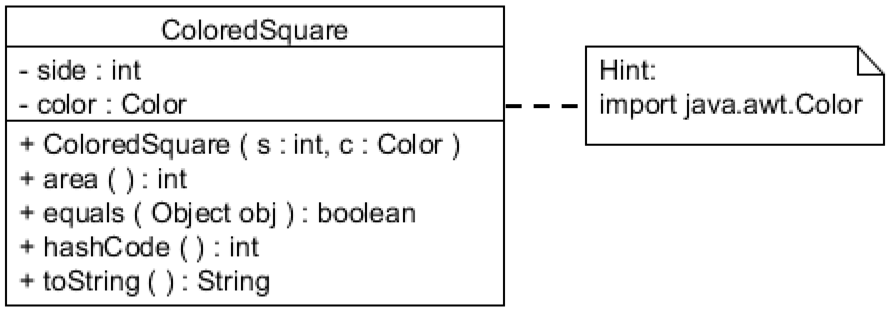
- Use the internet to learn about Set, HashSet, and other topics
- Understand the 2 major differences between interface Set and interface List
- Create, initialize, and display a HashSet
- Practice implementing a GUI application with WindowBuilder
- Experience the benefit of separating functionality from display

Description:

Download the starter project and import it into Eclipse.
Implement the classes ColoredSquare and ListVsSetDemo as described below. Use the console app provided to test your classes.
If the program doesn't work as expected change your code and **not** the console app.
Then modify the gui Application as described below so that it demonstrates the differences between the interfaces List and Set.

Ad ColoredSquare:

- Implement ColoredSquare as specified in the UML diagram. Do not add or remove any class members.
  Important: ColoredSquare should **NOT** include any print methods



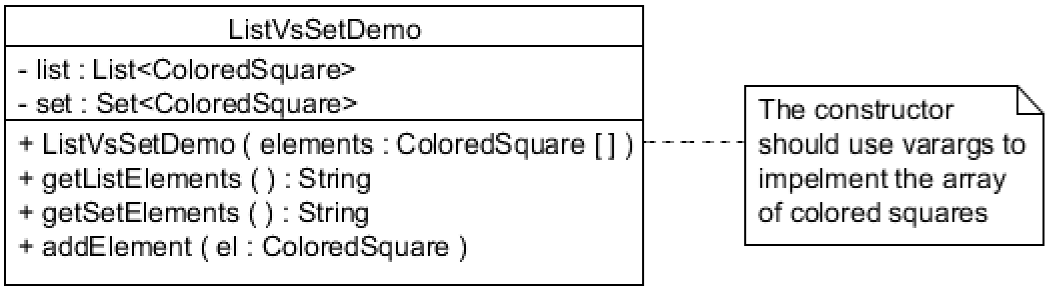
- **equals:**
  2 colored squares should be considered equal if they have the same size and color. **Use Eclipse to auto-implement equals and hashCode**
- **toString:**
  Eclipse can atuo-implement toString but in our case that would be a rather verbose description. Because of that I want you to write your own implementation that produces an output as shown on the right.
  e.g. `side:14 #0000FF   side:12 #FFFF00   etc.`
  Notice: the toString method of Color displays the rgb values in decimal format, however, in our implementation we use a hexadecimal format like in HTML.
  Hint 1: To get the individual red, green, blue values check out class Color .
  Hint 2: Info on how to format an integer as a 2-digit hexadecimal value

Console Output:

```
List:
side:14 #0000FF
side:18 #FF0000
side:12 #FFFF00
side:18 #FF0000
side:16 #00FF00

Set:
side:14 #0000FF
side:12 #FFFF00
side:16 #00FF00
side:18 #FF0000


Adding a new element:
List:
side:14 #0000FF
side:18 #FF0000
side:12 #FFFF00
side:18 #FF0000
side:16 #00FF00
side:10 #FFC800

Set:
side:14 #0000FF
side:12 #FFFF00
side:10 #FFC800
side:16 #00FF00
side:18 #FF0000


Adding a duplicate
element:
List:
side:14 #0000FF
side:18 #FF0000
side:12 #FFFF00
side:18 #FF0000
side:16 #00FF00
side:10 #FFC800
side:10 #FFC800

Set:
side:14 #0000FF
side:12 #FFFF00
side:10 #FFC800
side:16 #00FF00
side:18 #FF0000
```

Ad ListVsSetDemo:

- Implement ListVsSetDemo as specified in the UML diagram.
  Important: ListVsSetDemo should **NOT** include any print methods


ListVsSetDemo
- list : List<ColoredSquare>
- set : Set<ColoredSquare>
+ ListVsSetDemo (elements : ColoredSquare [ ])
+ getListElements ( ) : String
+ getSetElements ( ) : String
+ addElement (el : ColoredSquare)
The constructor should use varargs to impelment the array of colored squares

- Notice: even though the parameter of the constructor is defined as an array of type ColoredSquare, I want you to implement it as varargs. This is a good review and it provides a convenient way to call the constructor. Here is a short video in case you'd like a refresher on varargs:

- **constructor**:
  The constructor uses the elements passed to initialize both the list and the set

- **getListElements, getSetElements:**
  return a formatted string containing all the elements of the collection - one element per line. (see Console Output)
  Hint: use StringBuilder to create that string

- **addElement:**
  Adds the element passed to both the list and the set

Ad ListVsSetGuiApp:

Just as the console app calls methods of class ListVsSetDemo I want you to call methods of ListVsSetDemo to generate the text that will be displayed in ListVsSetGui when the ***Demo*** tab is selected.
The starter project includes two radio buttons: one to display the list elements the other to display the set elements.

Add a third radio button that allows adding a new element. (Just like in the console app it needs to be an element that has not been included in the original list. Every time the third radio button is selected the same 'new element' is added). Make sure to let the user know that an element got added.
If one of the first two radio buttons is selected after adding an element the updated list / set should be displayed.

When the user selects the ***ListVsSet*** menu item the two main differences between the interfaces List and Set should be displayed. No control panel is needed in this situation.

When the user selects the ***Exit*** menu item, the application should close.

Turning in:

Ensure to include your name on top of each java file that you modified.
Then zip up all four java files and turn them in via Canvas