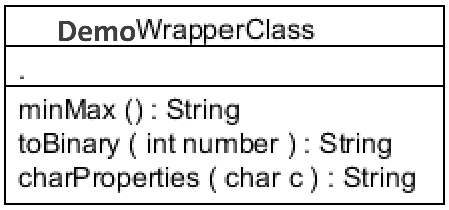
Lab Wrapper Class GUI

Description:

Download the starter project labWrapperClassGui.zip.
Unzip the files and import them into Eclipse. Check the *create top-level folder* check box so that the files are stored in their own folder.

In DemoWrapperClass.java do the following:

- Create 3 methods as specified in the UML class diagram:


DemoWrapperClass
minMax () : String
toBinary (int number) : String
charProperties (char c) : String

The methods need to use information and functionality provided by wrapper classes to produce the output described on your right

Ad charProperties:
This method checks whether the character passed is a white space, a digit, or a letter.
Notice: if the character passed happens to be a letter, the string doesn't display true but the original letter the vertical bar (|) and the same letter in the opposite case (see output)

In DemoWrapperClassConsole.java do the following:

- Call the 3 methods and print the results.
  For toBinary pass the number 15 hard coded
  For charProperties pass the character 'B' hard coded
  Each method-call should be preceded with a label (e.g. Method minMax:)

In DemoWrapperClassGui.java do the following:

- In Package Explorer right click DemoWrapperClassGui > Open With > WindowBuilder Editor
- Run

Ad Numbers

- Change the inactive Button to a radiobutton
  You can do that by right-clicking the JButton > Morph > jRadioButton
- Name the radio button ***Min Max***
- Rename the button (the name should be descriptive and fitting)
- Add a second radio button. It should be displayed right below the Min Max radio button. Name the second radio button ***Binary Oct Hex***
  Make sure to choose a descriptive name
- Run and click the radio buttons.
  Both radio buttons can be selected at the same time. This is not the way radio buttons are expected to work.

Output:

```
Method minMax:

Byte:
Min: -128
Max: 127

Short:
Min: -32768
Max: 32767

Integer:
Min: -2147483648
Max: 2147483647

Long:
Min: -9223372036854775808
Max: 9223372036854775807


Method toBinary:

Binary: 1111
Octal : 17
Hex   : f


Method charProperties:

White space: true
Digit: false
Letter: false

White space: false
Digit: true
Letter: false

White space: false
Digit: false
Letter: B|b

White space: false
Digit: false
Letter: b|B
```

In order to make the selection exclusive we need to tell Swing that both radio buttons belong to the same group.

- In design view select both radio buttons at the same time (hold the Ctrl key while clicking the second radio button) > right click > Set ButtonGroup > New standard
- Run again. This time the radio buttons are exclusive
- Add an ActionListener to each of the radio buttons
- When the Min Max button is clicked the min and max values of byte, short, int, and long should be displayed.
  This functionality is already avaialable in DemoWrapperClass. All we need is an instance of that class so we can call the appropriate method.
- Inside createNumberControlPanel declare and initialize a local variable of type DemoWrapperClass.
- Inside the event handlers (ActionListener) use this newly created instance of DemoWrapperClass to generate the text that should be displayed. Assign the text to the text property of the JTextArea (Hint: use the method setText)
- Run. At this point the text in the JTextArea on the right should change depending on the radio button that was selected.
  Also switch to the Char menu item. At this point it has a submit button on the left and a big label on the right. The first time you click the button the label turns blue. After that it no longer changes the color.

Ad Char:

- Now we want to modify the gui that is displayed when the Char menu item is chosen. However, the Design view still shows the gui we used for numbers. Let's change that.
- In Source view go to the method createContentPane.
  Currently the NumberControlPanel and the numberTextArea are added to the contentPane while the 2 lines of code, that would add charControlPanel and charLabel, are commented out.
  Reverse that. Uncomment the statements that add char related gui and comment out the two lines that added number related gui.
  Run – just to confirm that everything still works as expected
- In design view change the JLabel on the right side to a JTextArea. That is very analogous to changing a JButton to a JRadioButton (right click > Morph), however, the JTextArea is none of the choices displayed. You need to choose ***other***. A new window opens and allows you to choose a type (JTextArea).
  Make sure to refactor the field and method names.
- Add a margin of 30 to the JTextArea
- Run
- In design view change the jButton on the left side to a JTextField. Remember to refactor the name.
- Add an event handler (ActionListener) to the text field.
  Whenever the user presses enter the text of the text field should be displayed in the text area.
- Run. Notice how the width of the JTextField resizes after the user presses enter.
- Set the **preferred**Width of the charControlPanel to 150. Leave the height unchanged
- Run again. This time the JTextField no longer resizes with the input
- Change the ActionListener so that it displays the string produced by the method charProperties from DemoWrapperClass. The character passed to the method should be the first letter of the text in jTextField.
- Run