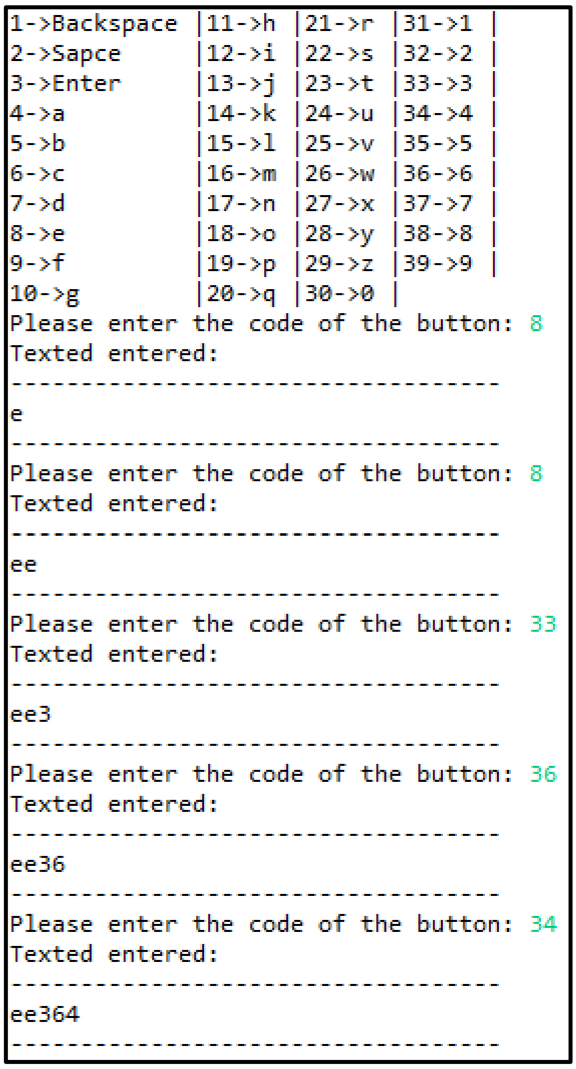
Inheritance

Objective:

The objective from this lab is to understand the importance of the inheritance concept in an object-oriented language (Java).

Part#1 keyboard simulator:

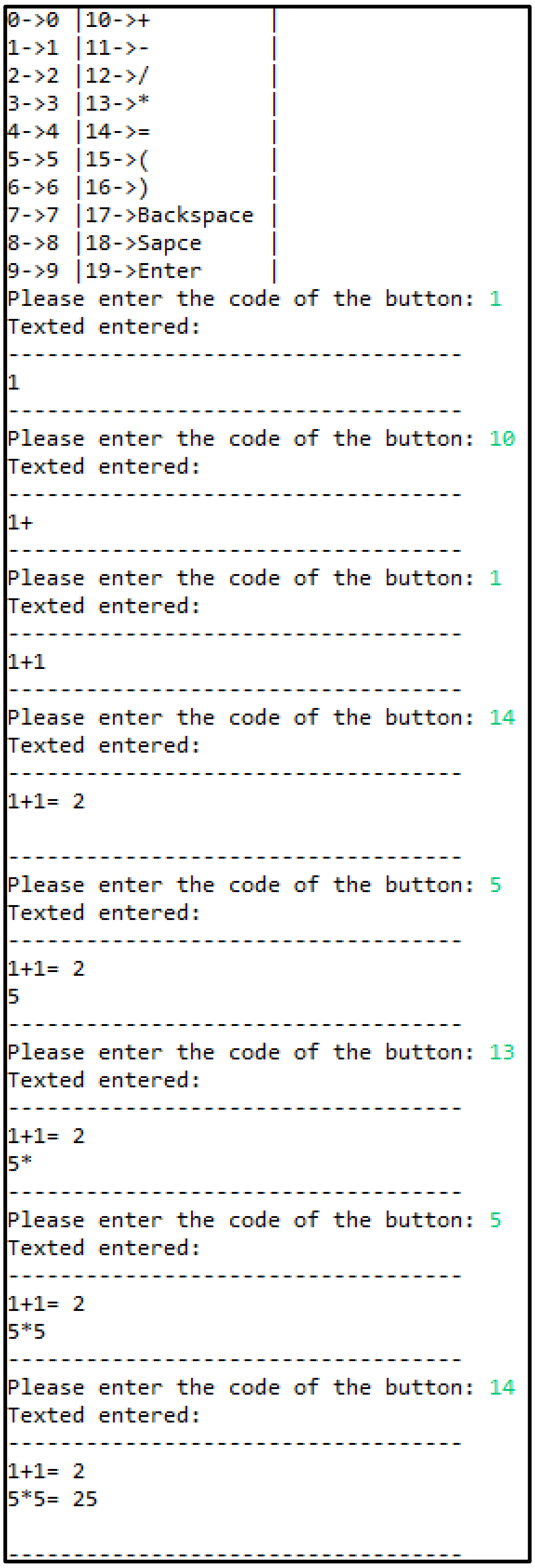
This lab is one of many labs that handle the creation of the keyboard simulator program. As you know, all keyboards talk to the computer using zeros and ones, which is not what we are going to do in this lab. This means that there is a special code for every button you press on the keyboard. So, the simulator main loop is pressing a button and generating a series of codes that the computer is translates into a value we see on the screen. Therefore, if we know the code of all the characters, we want the computer to show on screen, we can mimic the code sending to computer and the characters printing on the screen. In fact, we will give codes to all the characters, the ones we need, and if we input the right code, the keyboard simulator should print the corresponding character on the screen. Here is a sample of what the output should look like.



```
1->Backspace |11->h |21->r |31->1 |
2->Sapce     |12->i |22->s |32->2 |
3->Enter     |13->j |23->t |33->3 |
4->a         |14->k |24->u |34->4 |
5->b         |15->l |25->v |35->5 |
6->c         |16->m |26->w |36->6 |
7->d         |17->n |27->x |37->7 |
8->e         |18->o |28->y |38->8 |
9->f         |19->p |29->z |39->9 |
10->g        |20->q |30->0 |
Please enter the code of the button: 8
Texted entered:
-----------------------------------
e
-----------------------------------
Please enter the code of the button: 8
Texted entered:
-----------------------------------
ee
-----------------------------------
Please enter the code of the button: 33
Texted entered:
-----------------------------------
ee3
-----------------------------------
Please enter the code of the button: 36
Texted entered:
-----------------------------------
ee36
-----------------------------------
Please enter the code of the button: 34
Texted entered:
-----------------------------------
ee364
-----------------------------------
```

As you can see in the previous example the program starts with the table that shows all the characters in our keyboard (QWERTY keyboard) and the code for each one. After that, the program waits for the user to enter the code of the selected character and print the corresponding character between two the dash-lines. Every input is added in one place (displayedText).

As in every keyboard, there are 3 general buttons (Backspace, Space, Enter). Therefore, all the keyboards you are going to write in this lab should have these general buttons. In this lab, you shall write two classes that mimic the actions of two keyboard (QWERTY keyboard and Calculator keyboard). The buttons and the codes of the QWERTY keyboard is shown in the above output sample. The calculator keyboard output should look like the sample on the right.



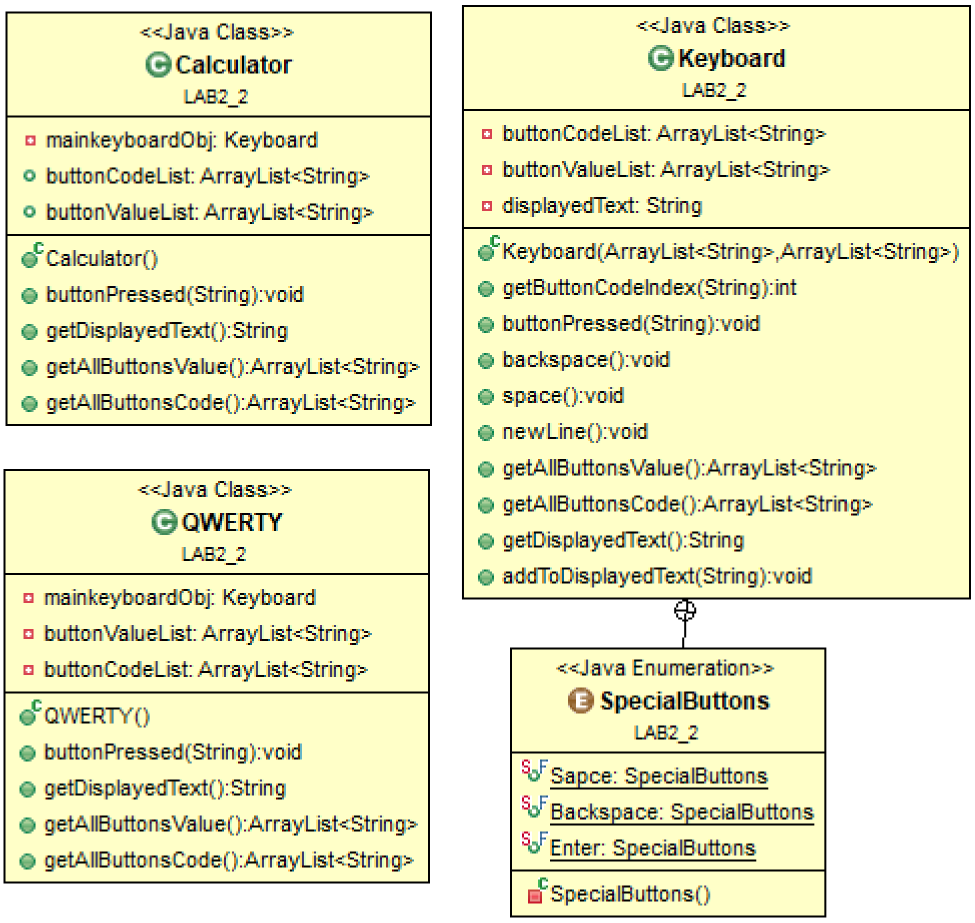
```
0->0 |10->+         |
1->1 |11->-         |
2->2 |12->/         |
3->3 |13->*         |
4->4 |14->=         |
5->5 |15->(         |
6->6 |16->)         |
7->7 |17->Backspace |
8->8 |18->Sapce     |
9->9 |19->Enter     |
Please enter the code of the button: 1
Texted entered:
-----------------------------------
1
-----------------------------------
Please enter the code of the button: 10
Texted entered:
-----------------------------------
1+
-----------------------------------
Please enter the code of the button: 1
Texted entered:
-----------------------------------
1+1
-----------------------------------
Please enter the code of the button: 14
Texted entered:
-----------------------------------
1+1= 2

-----------------------------------
Please enter the code of the button: 5
Texted entered:
-----------------------------------
1+1= 2
5
-----------------------------------
Please enter the code of the button: 13
Texted entered:
-----------------------------------
1+1= 2
5*
-----------------------------------
Please enter the code of the button: 5
Texted entered:
-----------------------------------
1+1= 2
5*5
-----------------------------------
Please enter the code of the button: 14
Texted entered:
-----------------------------------
1+1= 2
5*5= 25

-----------------------------------
```

The calculator keyboard acts a bit different than the QWERTY keyboard. The codes from 0 to 9 are reserved for numbers from 0 to 9. The user input the expression, one by one, and the calculator does nothing until the user enters the equal sign (=). Then the calculator evaluates the expression, prints the result next to the expression and jumps to a new line. The story does not end here! The user can enter another expression in the same way and ask for the results. The calculator is smart enough to distinguish the current expression from the previous one. They should not be mixed.

Since both keyboards should store two arrays, one for the codes list and one for the values list, there should be one object that holds the data for both of them and acts relatively the same. This object is an instance of the Keyboard class. This instance can store the two lists, act with the pressed buttons, add new character to the stored text (called displayedText in keyboard class), and return the two lists.

The action of pressing a button is simply accepting the entered code, looking for the code in the code list, getting the corresponding value of the button from the characters list and adding the new character to the displayed text. This action is forwarded from the two class (QWERTY keyboard and Calculator keyboard). However, the button pressed method in the calculator class checks if the entered value is the equal sign (=). If so, then it evaluates the expression using the evaluate method in the EvaluateString class (one of the provided classed). Finally, the button pressed method adds the result to the displayedText in the keyboard class.

The constructor of the two keyboards (QWERTY keyboard and Calculator keyboard) should create the two arrays with fixed characters (as shown in the sample output). Then creates the instance from the Keyboard class that accept those to lists and store them. In addition, the keyboard class constructor should generate the general buttons and add them to the list of buttons with their own codes. The value of the general buttons should be defined as an enumeration constant and used all over the program. Here is the UML diagram for all the classes.



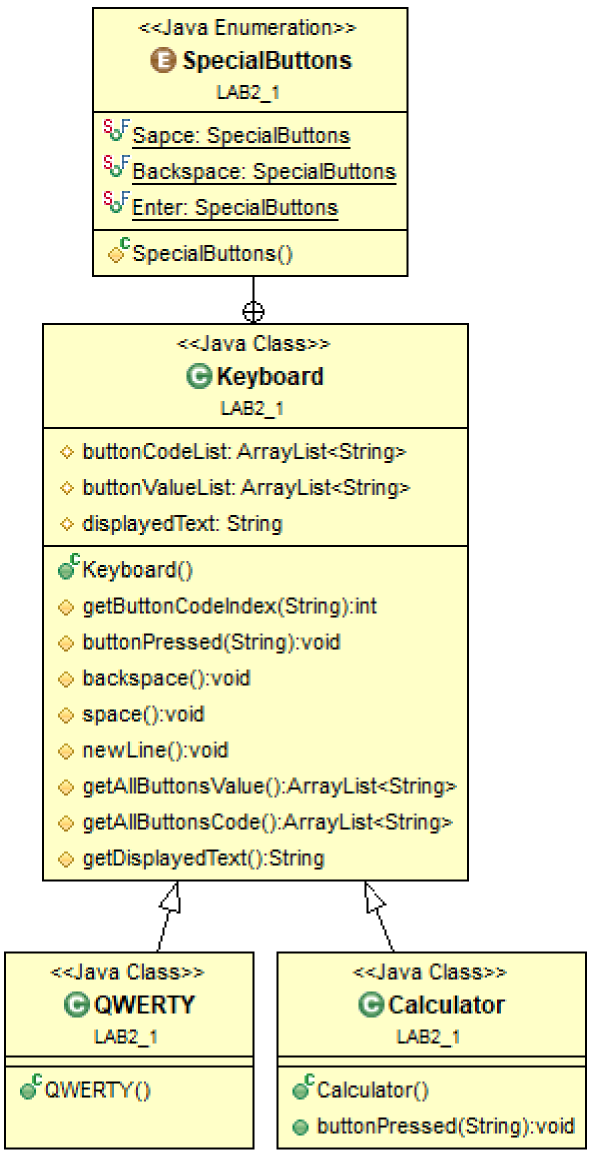
<<Java Class>>
Calculator
LAB2_2
mainkeyboardObj: Keyboard
buttonCodeList: ArrayList<String>
buttonValueList: ArrayList<String>
Calculator()
buttonPressed(String):void
getDisplayedText():String
getAllButtonsValue():ArrayList<String>
getAllButtonsCode():ArrayList<String>
<<Java Class>>
Keyboard
LAB2_2
buttonCodeList: ArrayList<String>
buttonValueList: ArrayList<String>
displayedText: String
Keyboard(ArrayList<String>,ArrayList<String>)
getButtonCodeIndex(String):int
buttonPressed(String):void
backspace():void
space():void
newLine():void
getAllButtonsValue():ArrayList<String>
getAllButtonsCode():ArrayList<String>
getDisplayedText():String
addToDisplayedText(String):void
<<Java Class>>
QWERTY
LAB2_2
mainkeyboardObj: Keyboard
buttonValueList: ArrayList<String>
buttonCodeList: ArrayList<String>
QWERTY()
buttonPressed(String):void
getDisplayedText():String
getAllButtonsValue():ArrayList<String>
getAllButtonsCode():ArrayList<String>
<<Java Enumeration>>
SpecialButtons
LAB2_2
Sapce: SpecialButtons
Backspace: SpecialButtons
Enter: SpecialButtons
SpecialButtons()

Note: You should use the TableViewer class, the EvaluationString class and the Testing class in your work for this lab.

Part#2 Using inheritance concept:

Know you should have a working keyboard simulator that works flawlessly. If that is the case then great job, otherwise you should finish part#1 first. Anyway, I am sure that you notice that we can use inheritance in this program. The QWERTY keyboard and the Calculator keyboard use all the features in the Keyboard class, so they should extend it. The keyboard class is the supertype for the other classes. In other words, the two keyboards (QWERTY keyboard and Calculator keyboard) are a special case of the main keyboard (the keyboard class). So, let us work on it.

To achieve this goal, first, the QWERTY keyboard and Calculator keyboard should be subclass of the keyboard class (by using the word extend). Second, you should work on removing all the unnecessary codes. For example, all the methods in the two subclasses that can be removed since they are all duplicates of the methods in the keyboard class. The button pressed methodin the Keyboard class can be overridden in the calculator class and uses the “super” keyword to call the one in the keyboard class. The following URL diagram should show all the differences from the previous one.



Note: You should **NOT** change the TableViewer class, the EvaluationString class and the Testing class.

Part#3 What is the difference?

As you can see in the output there is no difference between the output from the code in part#1 and the output from the code in part#2. The user shall use both programs without noticing any difference. The use doesn't know that you didn't use inheritance in part#1 nor using it in part#2. So, what is the difference? Why do we use inheritance? What are the advantages of using inheritance in this lab? Or in general?

What to turn in

You should turn your work in a **MS Word** file that has your codes for each part as specified below. All the codes should be yours. **NO COPY/PASTE IS ALLOWED**. You should always copy/paste the code from your compiler and take a screenshot of the console.

Part#1:

- The Keyboard class.
- The QWERTY class.
- The Calculator class.
- A Screenshot of the console after you run the main class that shows your testing.
- A txt file that has a copy of the testing of simulator output

Part#2:

- The Keyboard class.
- The QWERTY class.
- The Calculator class.
- A Screenshot of the console after you run the main class that shows your testing.
- A txt file that has a copy of the testing of simulator output

Part#3:

- Answer the question with examples.