Java Calculator

In this presentation, these things will be explained:

- Variables
- The use of functions
- About our GUI
- How the program works

Java Calculator - Variables

```
15 public class Test extends JFrame{
16     private JFrame frame;
17     private JTextArea textarea;
18     private JTextField textfield;
19     int phase = 0, a;
20     double number1 = 0, number2 = 0, result = 0;
21     String n1, n2, r, sys;
22     NumberFormat nf;
23     JButton num1;
24     int dothasdef = 0;
```

phase為階段變數
a則是表示運算子

Meanings of 'a'

+	1
-	2
X	3
/	4
SQRT	5

三個double為主要進行運算的變數
number1為使用者輸入的第一個變數
number2為使用者輸入的第二個變數
result即為運算結果

NumberFormat是要
統一格式而建立的變數

dothasdef是要分開使用
者是在哪些變數加上小數點
而設立的變數

這裡的字串都會是經由前面數字轉換顯示到螢幕上
n1, n2, r 分別為上面3個轉換的字串
sys則是要做長度檢查而設立的暫存字串

Java Calculator - Variables

```
778         // This will set number format (e.g. dump no needed zeros)
779         nf = new DecimalFormat("####.####");

789                     n1 = nf.format(number1);
790                     textfield.setText(n1);
```

DecimalFormat() can dump unnecessary zeros.

Phases (how the program works):

phase 0 : After starting the program, nothing defined.
phase 1 : Only number1 has been defined.
phase 2 : number1 with an operator ‘a’ defined.
phase 3 : Both number1 and number2 with an operator defined.
phase 33 : After phase 4, any operator will rewrite number2.
phase 4 : When equal button was pressed, result defined.

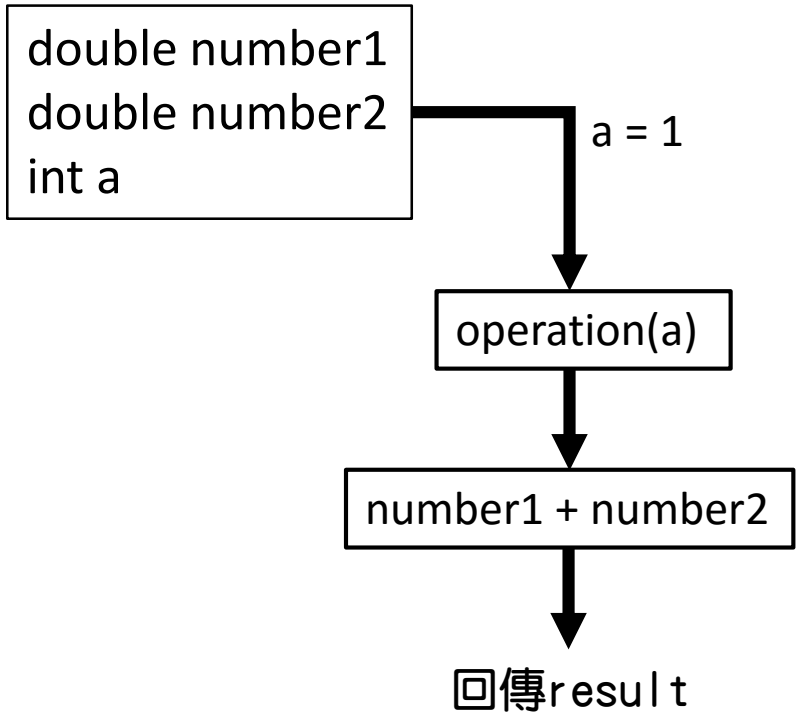
```
public void actionPerformed(ActionEvent e) {
    switch(phase){
    case 0:
        // at phase 0, number1 = 0
        number1 = 0;
        a = 3;
        phase = 2;
        break;
    case 1:
        // at phase 1, number1 and a will be de
        a = 3;
        // and the next phase, 2
        phase = 2;
        break;
    case 2:
        // Now, an operator has been defined, s
        a = 3;
        break;
    case 3:
        // instructions shows at plus button
        result = operation(3);
        number1 = number2 = result;
        r = nf.format(result);
        textfield.setText(r);
        a = 3;
        number1 = result;
        phase = 2;
        break;
    case 4:
        // Still feeling complicated? both numb
        // only rewrites operator -> goto phase
        number1 = result;
        number2 = result;
        n1 = nf.format(number1);
        textfield.setText(n1);
        a = 3;
        phase = 33;
        break;
    }
```

Java Calculator - Functions

```
1566      void clean(){
1567          // Slaughterhouse
1568          number1 = 0;
1569          number2 = 0;
1570          phase = 0;
1571          a = 0;
1572          textfield.setText("0");
1573      }
```

- clean() is made for clearing all variables
- Only number1, number2, phase, dothasdef, and a will be cleared
- Other string variables will be cleared every time we enter a number

Java Calculator - Functions

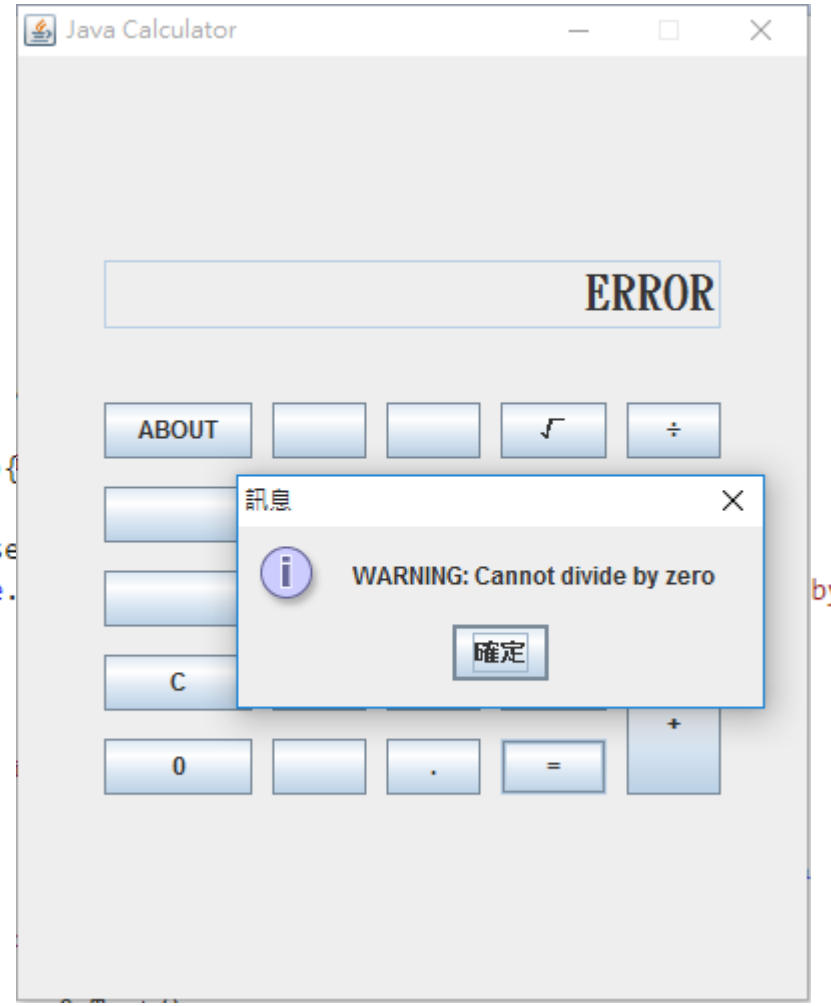



```
1512    public double operation(int a){
1513        double rtn = 0;
1514        if(a==1){
1515    +       // plus
1516            rtn = number1 + number2;
1517        }else if(a == 2){
1518    -       // minus
1519            rtn = number1 - number2;
1520        }else if(a == 3){
1521    *       rtn = number1 * number2;
1522        }else if(a == 4){
1523            /*
1524            try{
1525                rtn = number1/number2;
1526            }catch(ArithmeticException ex){
1527    /           textfield.setText("ERROR");
1528            }*/
1529            if(number2 == 0){
1530                erhandle();
1531            }else{
1532                rtn = number1/number2;
1533            }
1534        }else if(a == 5){
1535    √       rtn = Math.sqrt(number1);
1536        }
1537        return rtn;
1538    }
```

Java Calculator - Functions

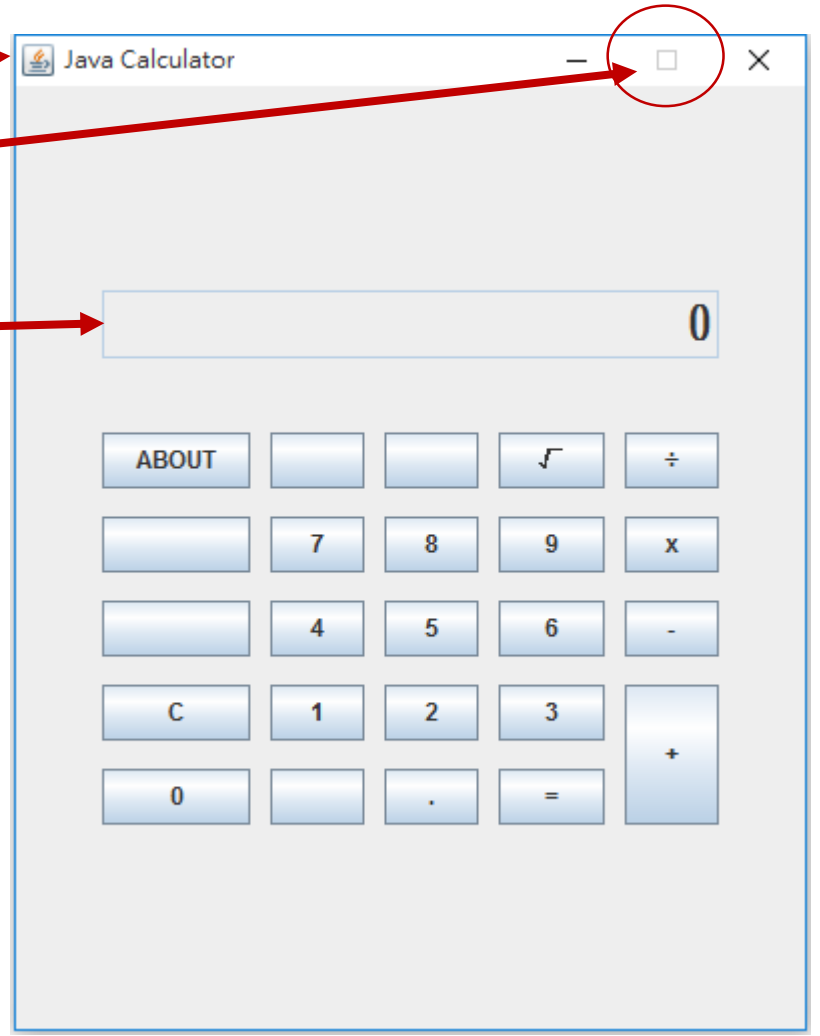
```
1512    public double operation(int a){
1513        double rtn = 0;
1514        if(a==1){
1515            // plus
1516            rtn = number1 + number2;
1517        }else if(a == 2){
1518            // minus
1519            rtn = number1 - number2;
1520        }else if(a == 3){
1521            rtn = number1 * number2;
1522        }else if(a == 4){
1523            /*
1524            try{
1525                rtn = number1/number2;
1526            }catch(ArithmeticException ex){
1527                textfield.setText("ERROR");
1528            }*/
1529            if(number2 == 0){
1530                erhandle();
1531            }else{
1532                rtn = number1/number2;
1533            }
1534        }else if(a == 5){
1535            rtn = Math.sqrt(number1);
1536        }
1537        return rtn;
1538    }
```

```
1575    void erhandle(){
1576        clean();
1577        textfield.se
1578        JOptionPane.                                    by zero");
1579    }
```



Java Calculator - GUI

```
28      public void Test(){
29          // TODO Auto-generated method stub
30          frame = new JFrame("Java Calculator");
31          frame.setBounds(400,100,400,500);
32          frame.setLayout(new GridBagLayout());
33          frame.setResizable(false);
34          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35
36          textfield = new JTextField(8);
37          textfield.setEditable(false);
38          textfield.setHorizontalAlignment(JTextField.RIGHT);
39          Font font = new Font("標楷體", Font.BOLD,24);
40          textfield.setFont(font);
41
42
43          GridBagConstraints t00 = new GridBagConstraints();
44          t00.gridx = 0;
45          t00.gridy = 0;
46          t00.gridwidth = 5;
47          t00.gridheight = 4;
48          t00.fill = GridBagConstraints.HORIZONTAL;
49          t00.anchor = GridBagConstraints.NORTH;
50          t00.insets = new Insets(7,5,30,5);
51
52          //frame.add(textarea,t00);
53          frame.add(textfield,t00);
54          textfield.setText("0");
```

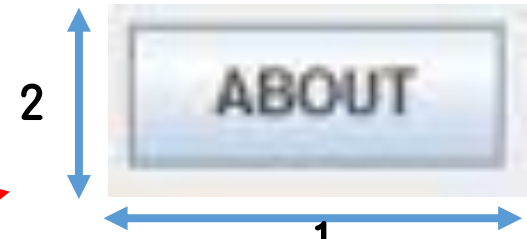


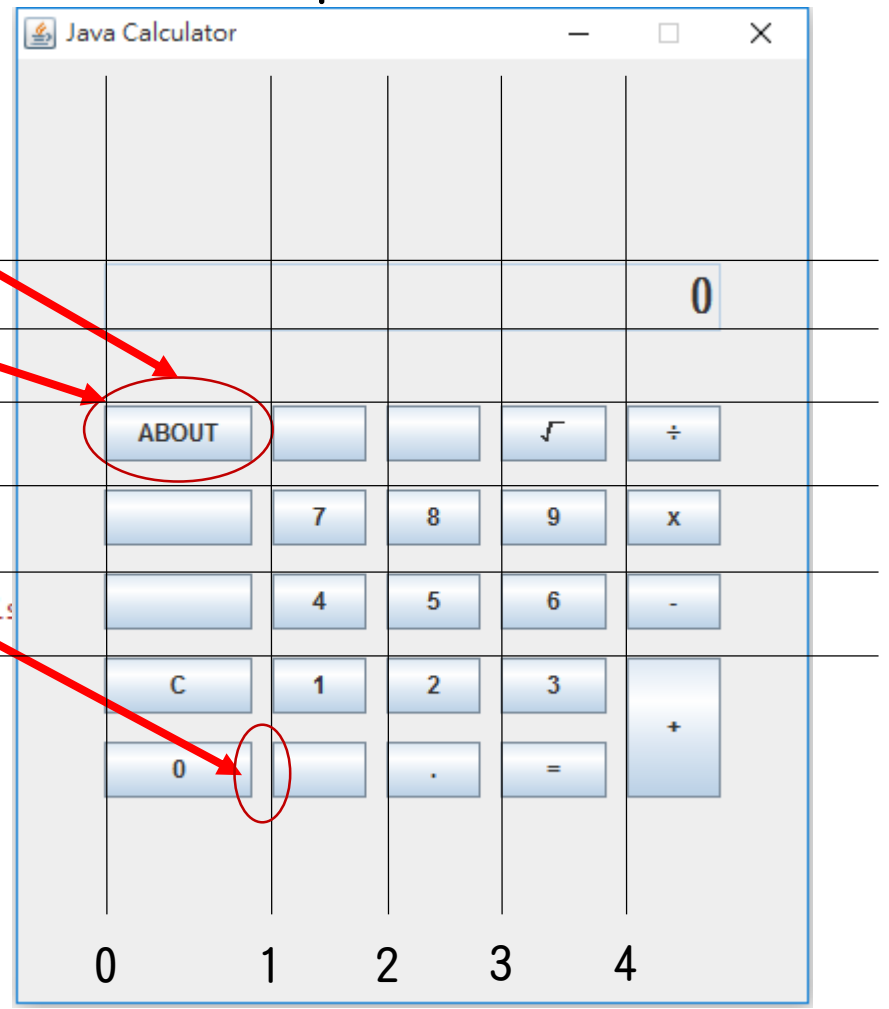
Java Calculator - GUI

```
56          JButton babout = new JButton("ABOUT");
57          GridBagConstraints b01 = new GridBagConstraints();
58          b01.gridx = 0;
59          b01.gridy = 4;
60          b01.gridwidth = 1;
61          b01.gridheight = 2;
62          b01.fill = GridBagConstraints.HORIZONTAL;
63          b01.anchor = GridBagConstraints.NORTH;
64          b01.insets = new Insets(7,5,7,5);
65
66          frame.add(babout,b01);


404     babout.addActionListener(new ActionListener(){  //about
405
406         public void actionPerformed(ActionEvent e) {
407             JOptionPane.showMessageDialog(null, "Made By hae0347\nFor more info, vis
408          }
409
410     });
```

(0, 4)





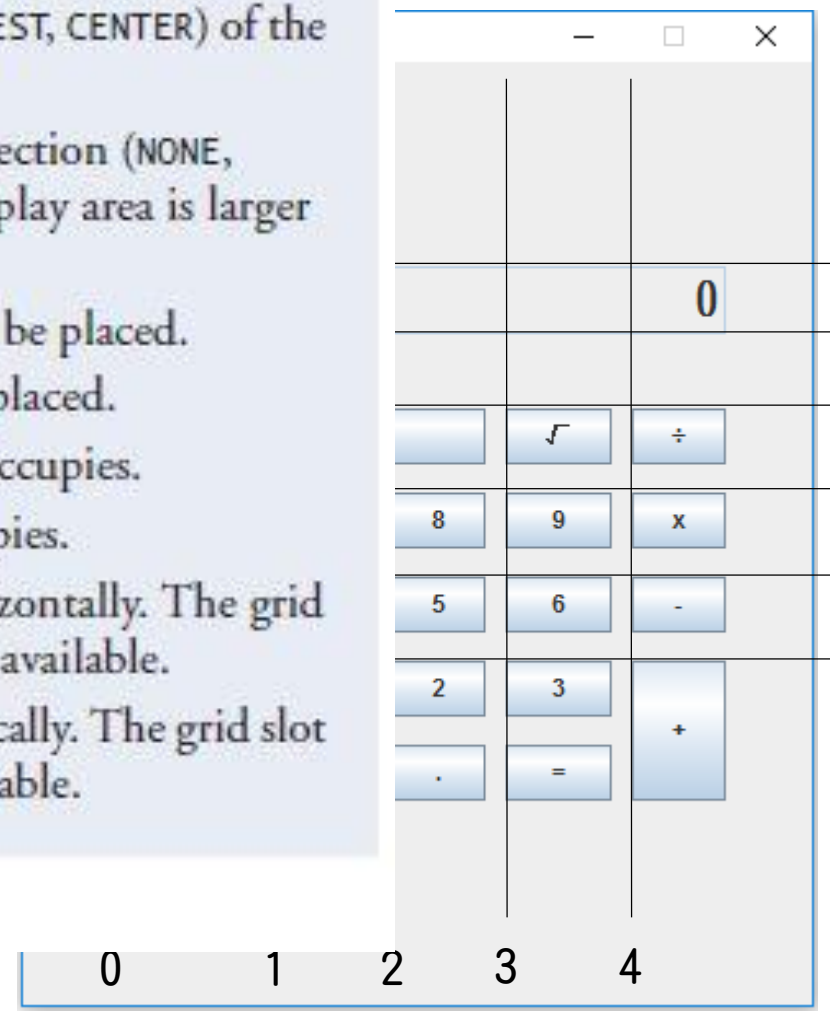
Java

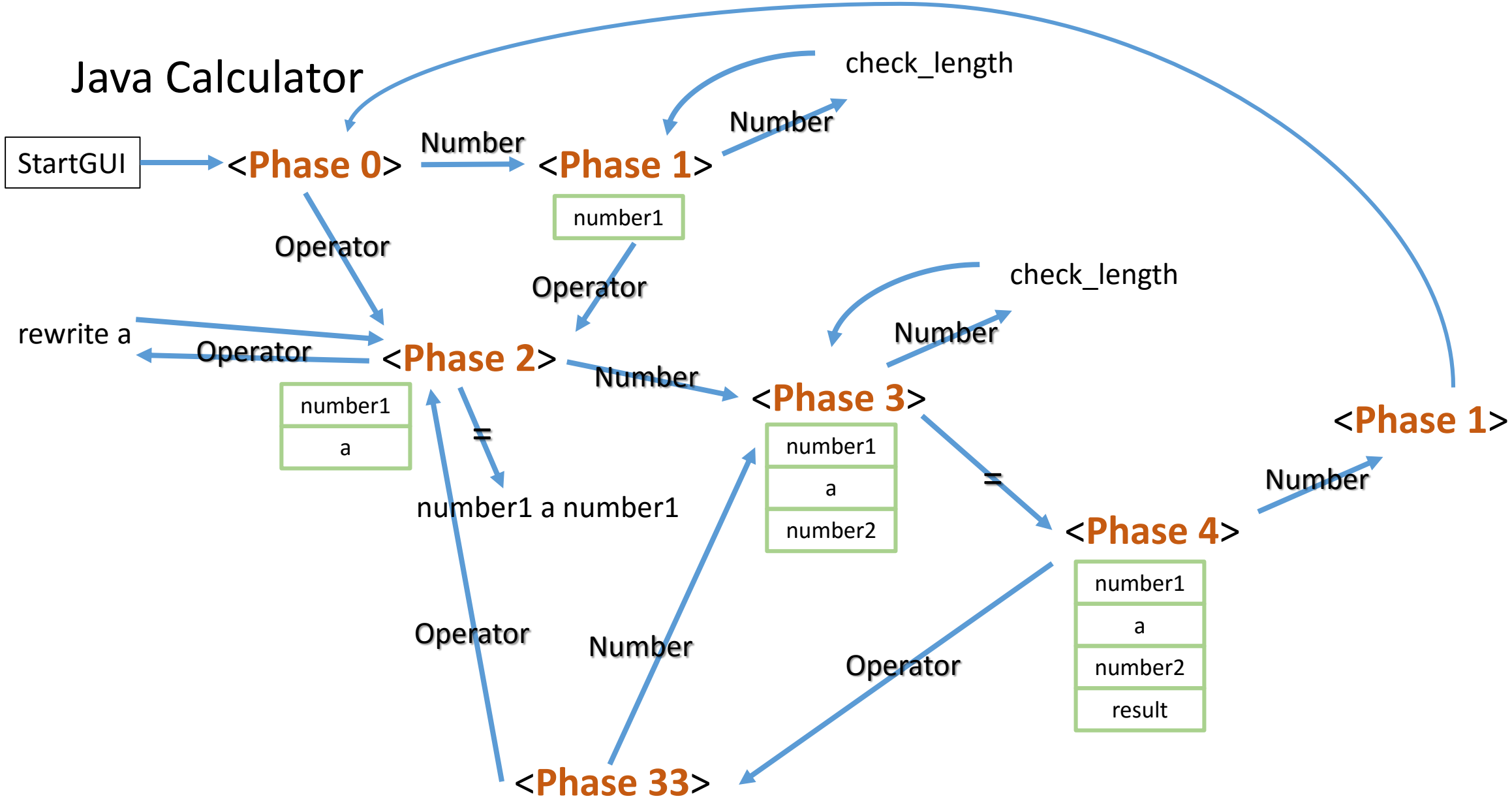
Field	Description
anchor	Specifies the relative position (NORTH, NORTHEAST, EAST, SOUTHEAST, SOUTH, SOUTHWEST, WEST, NORTHWEST, CENTER) of the component in an area that it does not fill.
fill	Resizes the component in the specified direction (NONE, HORIZONTAL, VERTICAL, BOTH) when the display area is larger than the component.
gridx	The column in which the component will be placed.
gridy	The row in which the component will be placed.
gridwidth	The number of columns the component occupies.
gridheight	The number of rows the component occupies.
weightx	The amount of extra space to allocate horizontally. The grid slot can become wider when extra space is available.
weighty	The amount of extra space to allocate vertically. The grid slot can become taller when extra space is available.

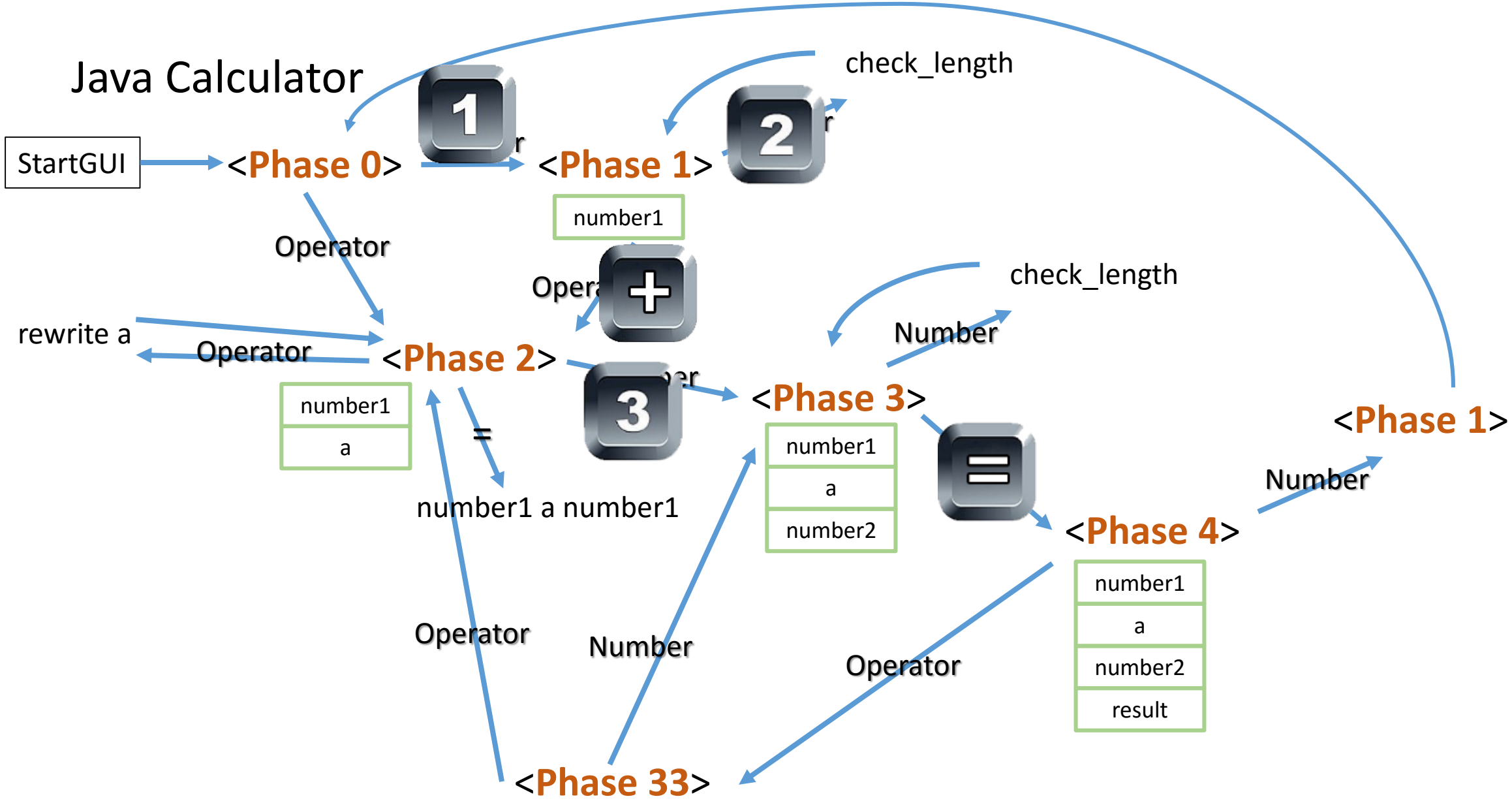
Fig. 25.19 | GridBagConstraints fields.

```
56
57
58
59
60
61
62
63
64
65
66

404     babou
405
406
407
408
409
410 });
```




Java Calculator
StartGUI
<Phase 0>
Number
<Phase 1>
check_length
Number
number1
Operator
Operator
rewrite a
Operator
<Phase 2>
number1
a
=
number1 a number1
Number
<Phase 3>
check_length
Number
number1
a
number2
=
<Phase 4>
number1
a
number2
result
Number
<Phase 1>
Operator
Number
Operator
<Phase 33>


Java Calculator
StartGUI
<Phase 0>
<Phase 1>
check_length
number1
Operator
rewrite a
Operator
<Phase 2>
number1
a
=
number1 a number1
<Phase 3>
check_length
Number
number1
a
number2
<Phase 4>
number1
a
number2
result
<Phase 1>
Number
Operator
Number
Operator
<Phase 33>