Java 计算器作业

2013599 田佳业

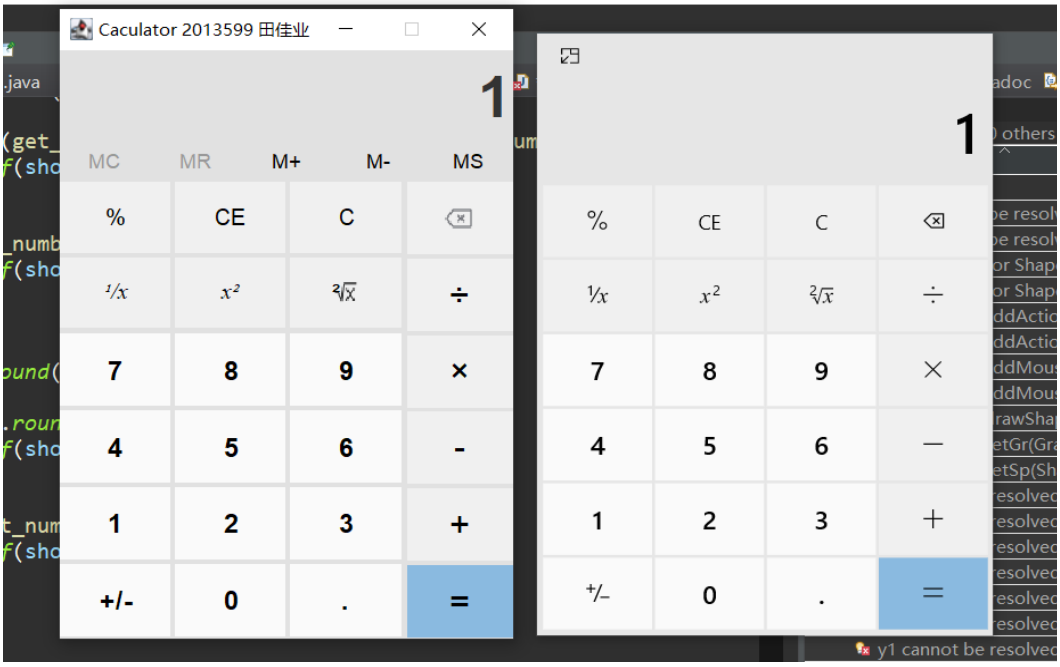
一、设计目标：

使用 JFrame 完成一个计算器界面。

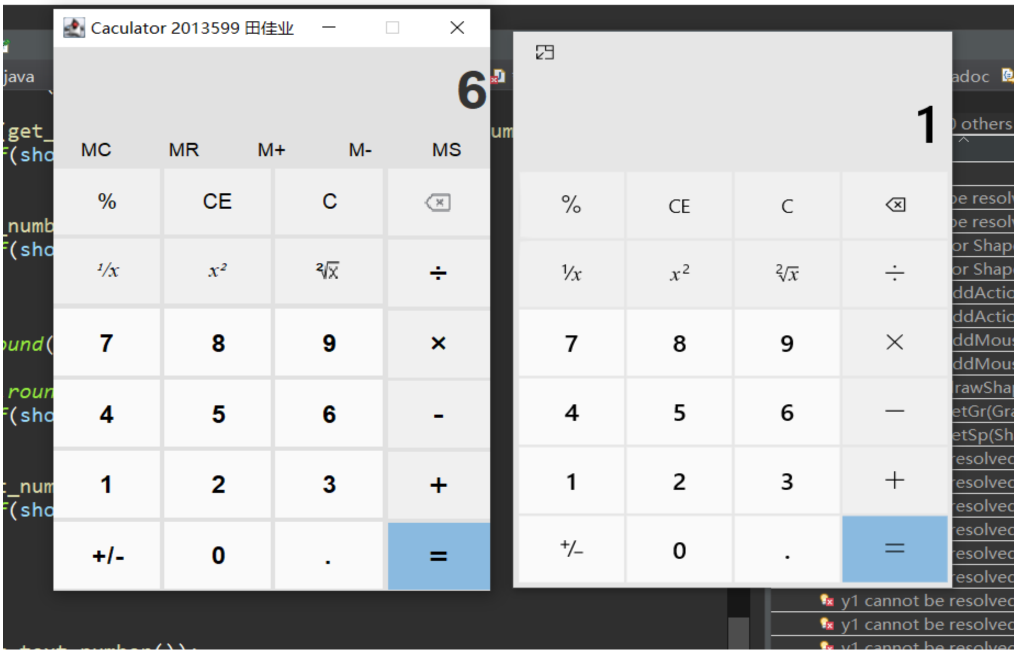
二、程序亮点：

1.完全真实的还原了 Win10 计算器的界面。
2.实现了面板上所有的运算，并额外添加记忆功能。
3.所有的运算行为均与 Win10 计算器一致。

三、运行实例：



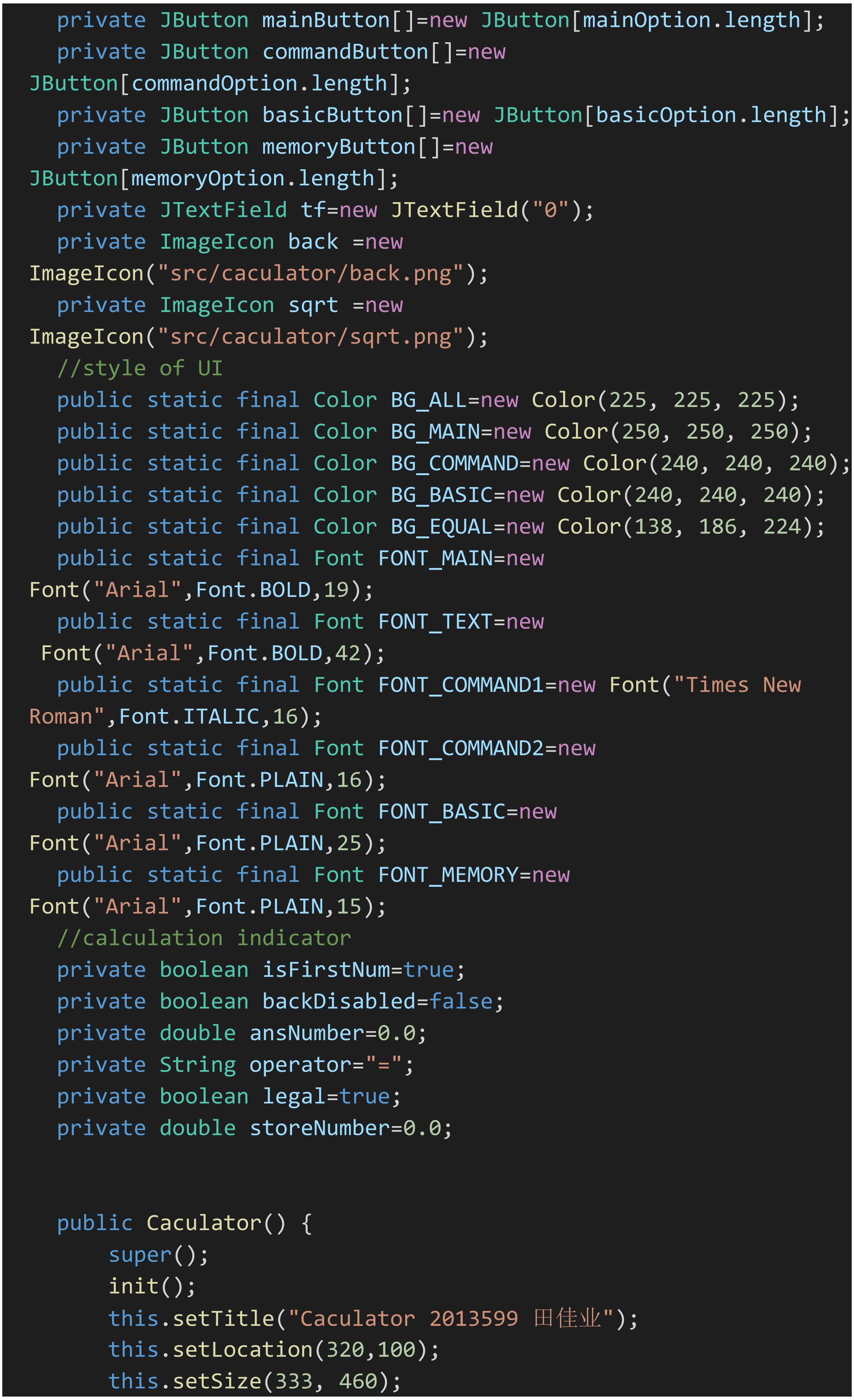
基本面板



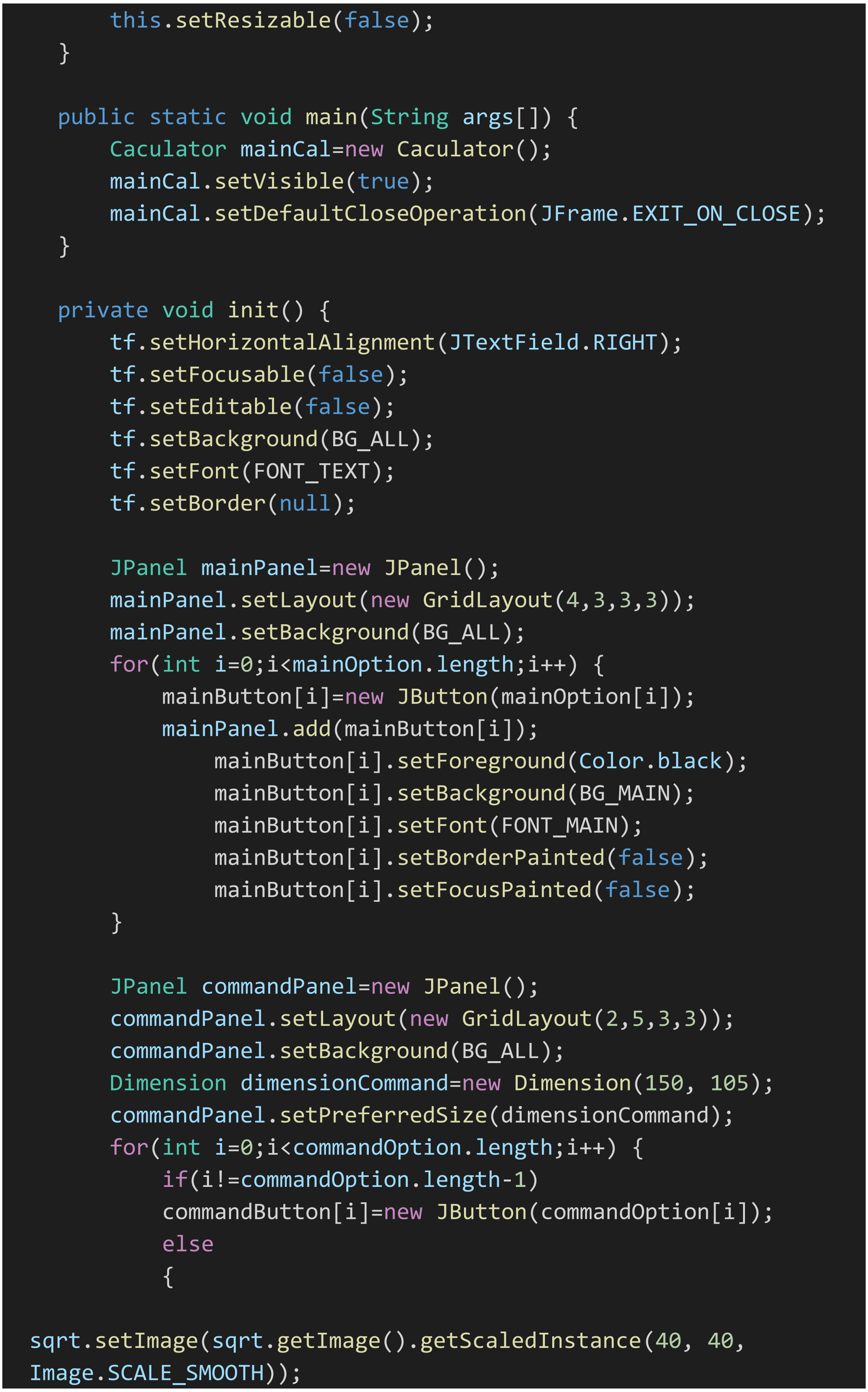
记忆功能

四、程序代码：

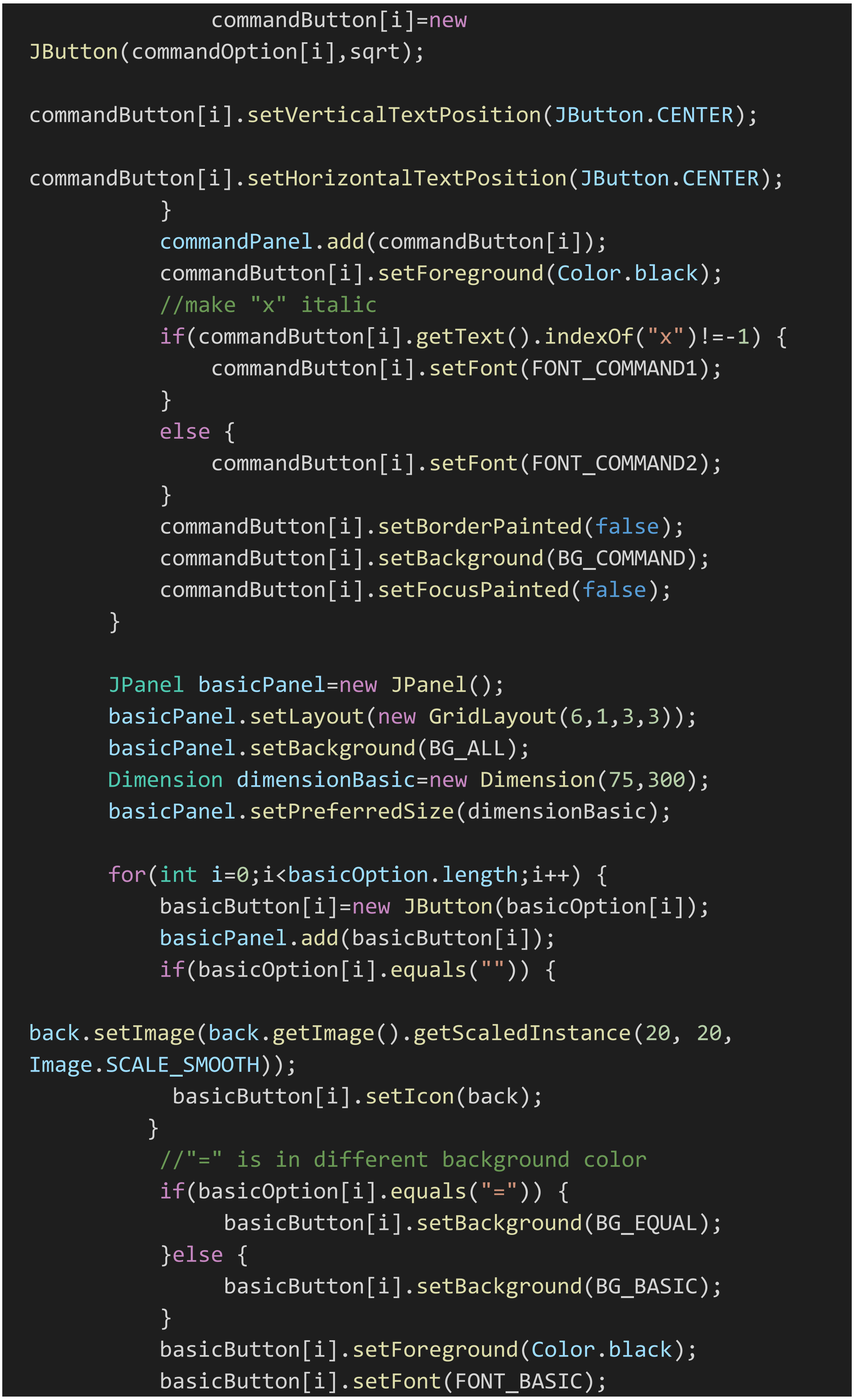
```
 package caculator;
 import java.awt.*;
 import java.awt.event.ActionEvent;
 import java.awt.event.ActionListener;
 import java.util.Arrays;

 import javax.swing.Icon;
 import javax.swing.ImageIcon;
 import javax.swing.JButton;
 import javax.swing.JFrame;
 import javax.swing.JPanel;
 import javax.swing.JTextField;

 public class Caculator extends JFrame implementsActionListener
{
   private static final long serialVersionUID = 1L;
   private final String[] mainOption={"7","8","9","4","5","6",
           "1","2","3","+/-","0","."};
   private final String[] memoryOption={"MC","MR","M+","M-
 ","MS"};
   private final String[]
 commandOption={"%","CE","C","¹/x","x²","²   "};
   private final String[] basicOption={"","÷","×","-","+","="};
```



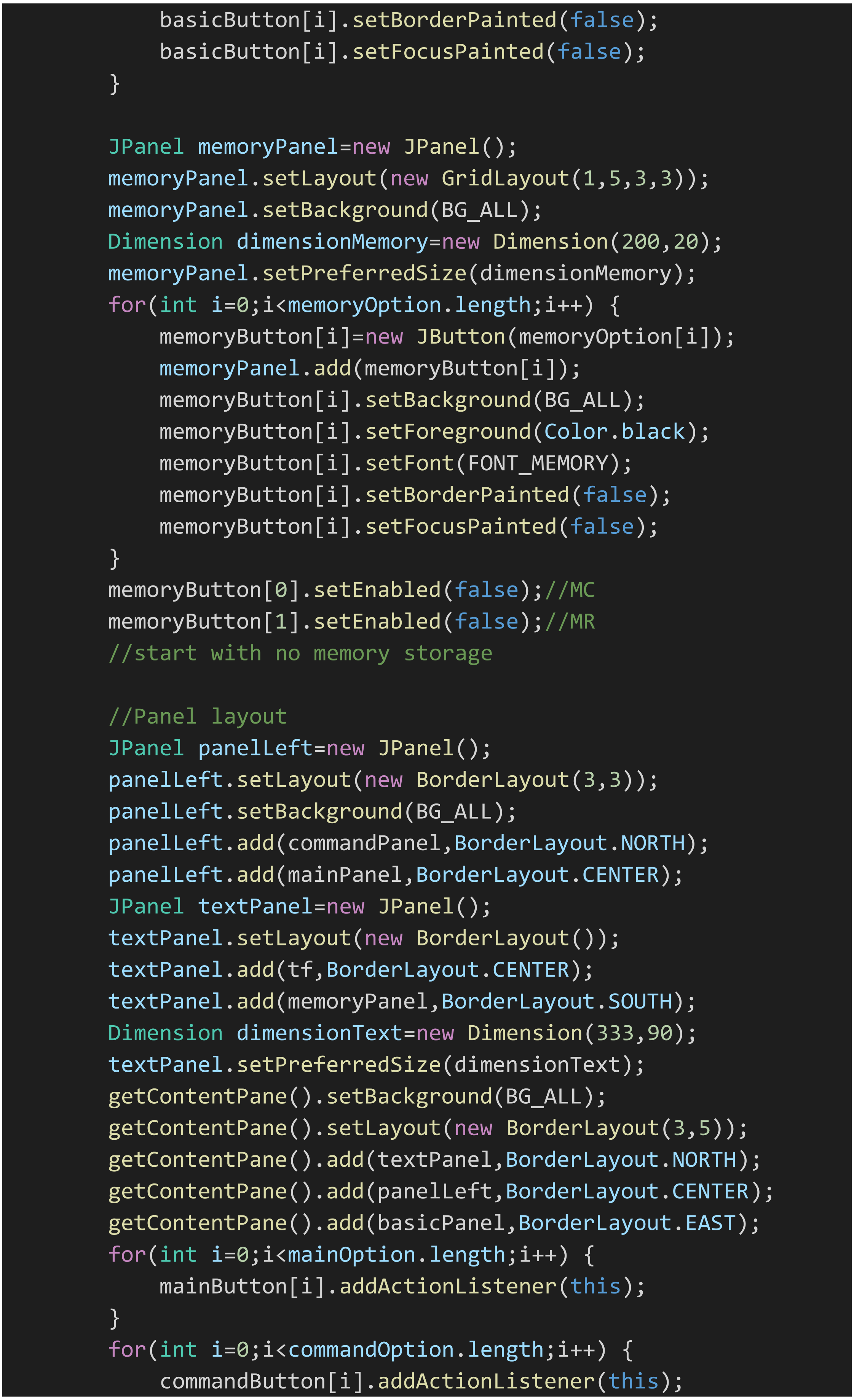
```
  private JButton mainButton[]=new JButton[mainOption.length];
  private JButton commandButton[]=new
JButton[commandOption.length];
  private JButton basicButton[]=new JButton[basicOption.length];
  private JButton memoryButton[]=new
JButton[memoryOption.length];
  private JTextField tf=new JTextField("0");
  private ImageIcon back =new
ImageIcon("src/caculator/back.png");
  private ImageIcon sqrt =new
ImageIcon("src/caculator/sqrt.png");
  //style of UI
  public static final Color BG_ALL=new Color(225, 225, 225);
  public static final Color BG_MAIN=new Color(250, 250, 250);
  public static final Color BG_COMMAND=new Color(240, 240, 240);
  public static final Color BG_BASIC=new Color(240, 240, 240);
  public static final Color BG_EQUAL=new Color(138, 186, 224);
  public static final Font FONT_MAIN=new
Font("Arial",Font.BOLD,19);
  public static final Font FONT_TEXT=new
 Font("Arial",Font.BOLD,42);
  public static final Font FONT_COMMAND1=new Font("Times New
Roman",Font.ITALIC,16);
  public static final Font FONT_COMMAND2=new
Font("Arial",Font.PLAIN,16);
  public static final Font FONT_BASIC=new
Font("Arial",Font.PLAIN,25);
  public static final Font FONT_MEMORY=new
Font("Arial",Font.PLAIN,15);
  //calculation indicator
  private boolean isFirstNum=true;
  private boolean backDisabled=false;
  private double ansNumber=0.0;
  private String operator="=";
  private boolean legal=true;
  private double storeNumber=0.0;


  public Caculator() {
      super();
      init();
      this.setTitle("Caculator 2013599 田佳业");
      this.setLocation(320,100);
      this.setSize(333, 460);
```



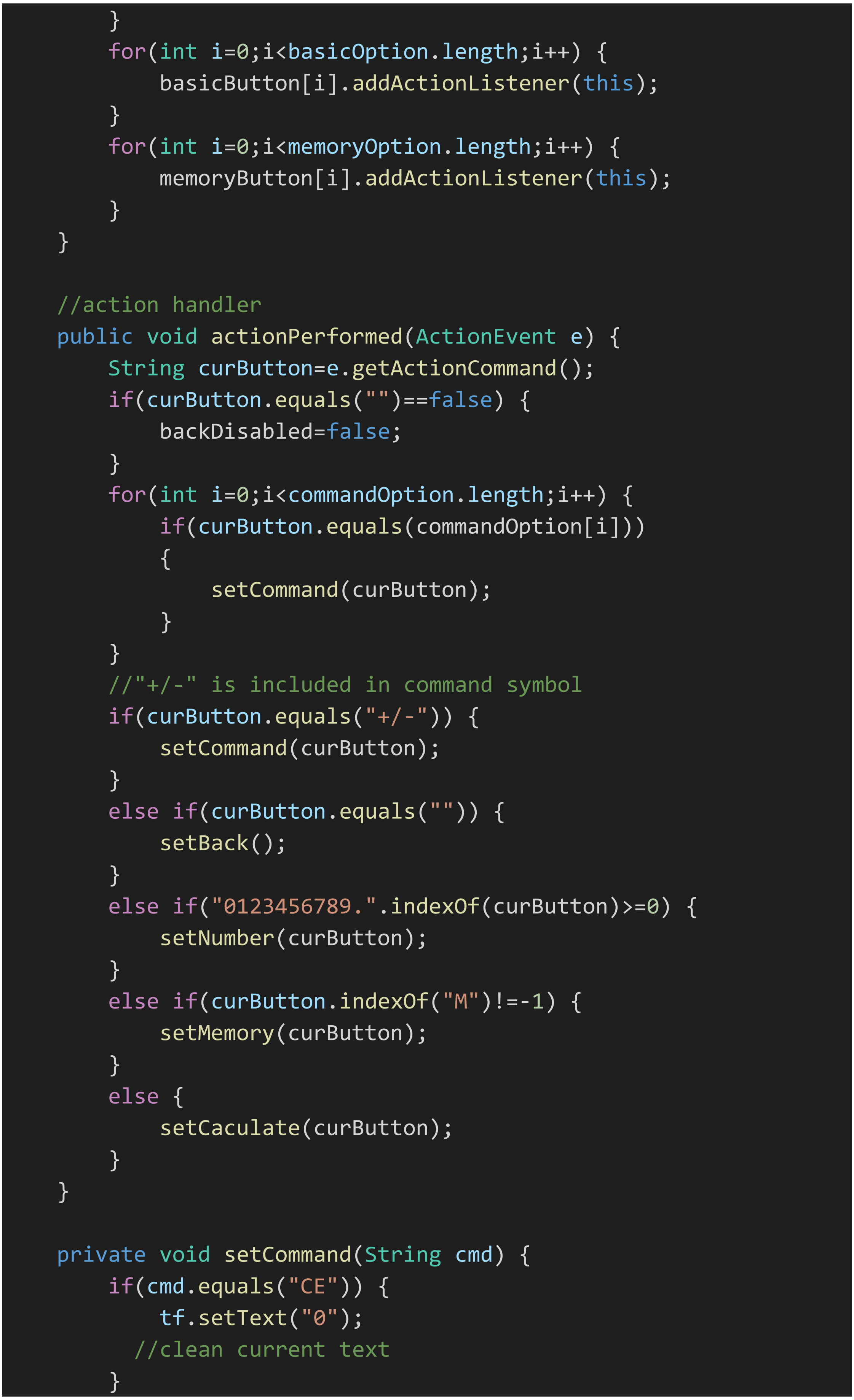
```
        this.setResizable(false);
    }

    public static void main(String args[]) {
        Caculator mainCal=new Caculator();
        mainCal.setVisible(true);
        mainCal.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    private void init() {
        tf.setHorizontalAlignment(JTextField.RIGHT);
        tf.setFocusable(false);
        tf.setEditable(false);
        tf.setBackground(BG_ALL);
        tf.setFont(FONT_TEXT);
        tf.setBorder(null);

        JPanel mainPanel=new JPanel();
        mainPanel.setLayout(new GridLayout(4,3,3,3));
        mainPanel.setBackground(BG_ALL);
        for(int i=0;i<mainOption.length;i++) {
            mainButton[i]=new JButton(mainOption[i]);
            mainPanel.add(mainButton[i]);
                mainButton[i].setForeground(Color.black);
                mainButton[i].setBackground(BG_MAIN);
                mainButton[i].setFont(FONT_MAIN);
                mainButton[i].setBorderPainted(false);
                mainButton[i].setFocusPainted(false);
        }

        JPanel commandPanel=new JPanel();
        commandPanel.setLayout(new GridLayout(2,5,3,3));
        commandPanel.setBackground(BG_ALL);
        Dimension dimensionCommand=new Dimension(150, 105);
        commandPanel.setPreferredSize(dimensionCommand);
        for(int i=0;i<commandOption.length;i++) {
            if(i!=commandOption.length-1)
            commandButton[i]=new JButton(commandOption[i]);
            else
            {

  sqrt.setImage(sqrt.getImage().getScaledInstance(40, 40,
  Image.SCALE_SMOOTH));
```



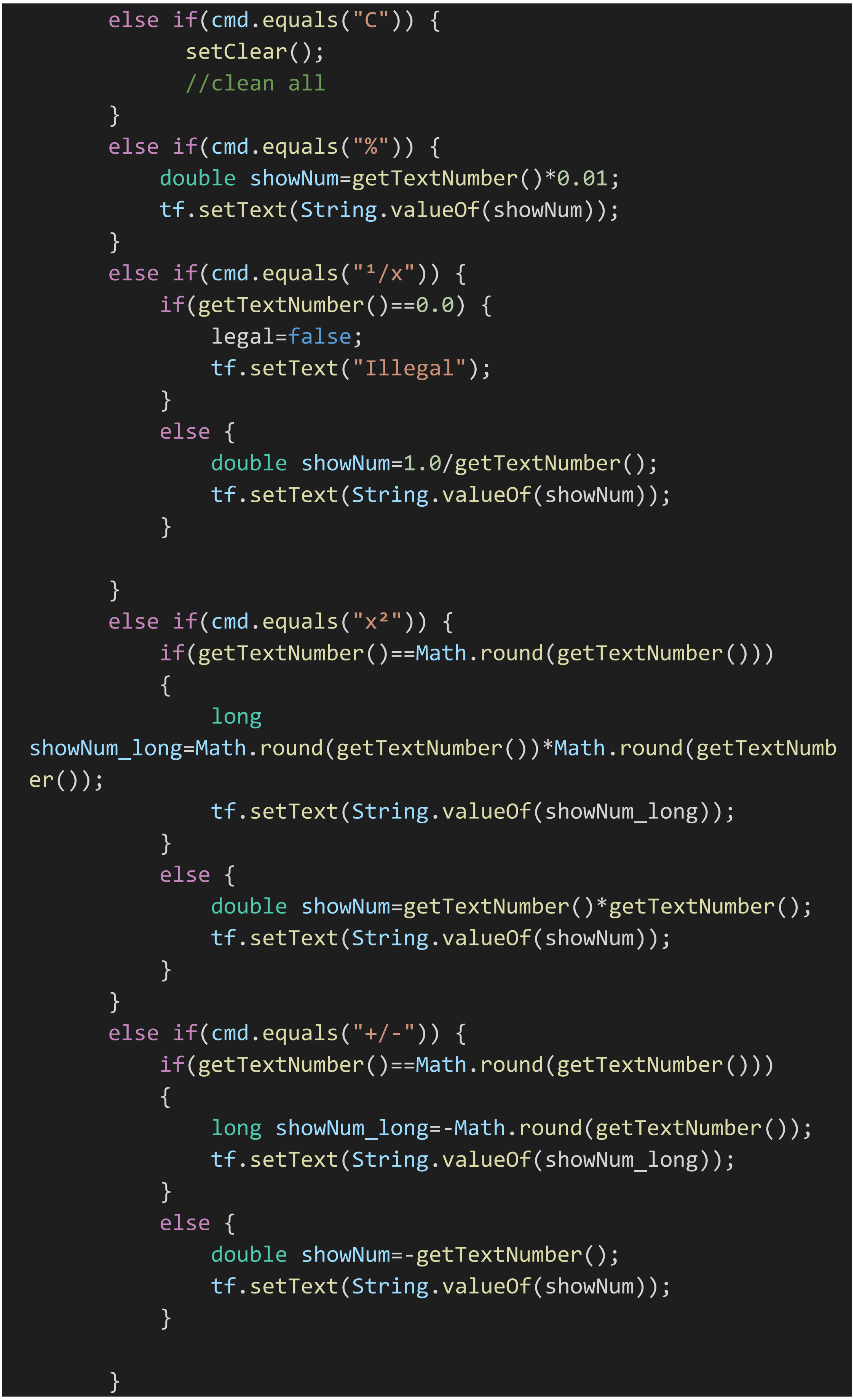
```
              commandButton[i]=new
JButton(commandOption[i],sqrt);

commandButton[i].setVerticalTextPosition(JButton.CENTER);

commandButton[i].setHorizontalTextPosition(JButton.CENTER);
          }
          commandPanel.add(commandButton[i]);
          commandButton[i].setForeground(Color.black);
          //make "x" italic
          if(commandButton[i].getText().indexOf("x")!=-1) {
              commandButton[i].setFont(FONT_COMMAND1);
          }
          else {
              commandButton[i].setFont(FONT_COMMAND2);
          }
          commandButton[i].setBorderPainted(false);
          commandButton[i].setBackground(BG_COMMAND);
          commandButton[i].setFocusPainted(false);
      }

      JPanel basicPanel=new JPanel();
      basicPanel.setLayout(new GridLayout(6,1,3,3));
      basicPanel.setBackground(BG_ALL);
      Dimension dimensionBasic=new Dimension(75,300);
      basicPanel.setPreferredSize(dimensionBasic);

      for(int i=0;i<basicOption.length;i++) {
          basicButton[i]=new JButton(basicOption[i]);
          basicPanel.add(basicButton[i]);
          if(basicOption[i].equals("")) {

back.setImage(back.getImage().getScaledInstance(20, 20,
Image.SCALE_SMOOTH));
           basicButton[i].setIcon(back);
         }
          //"=" is in different background color
          if(basicOption[i].equals("=")) {
               basicButton[i].setBackground(BG_EQUAL);
          }else {
               basicButton[i].setBackground(BG_BASIC);
          }
          basicButton[i].setForeground(Color.black);
          basicButton[i].setFont(FONT_BASIC);
```



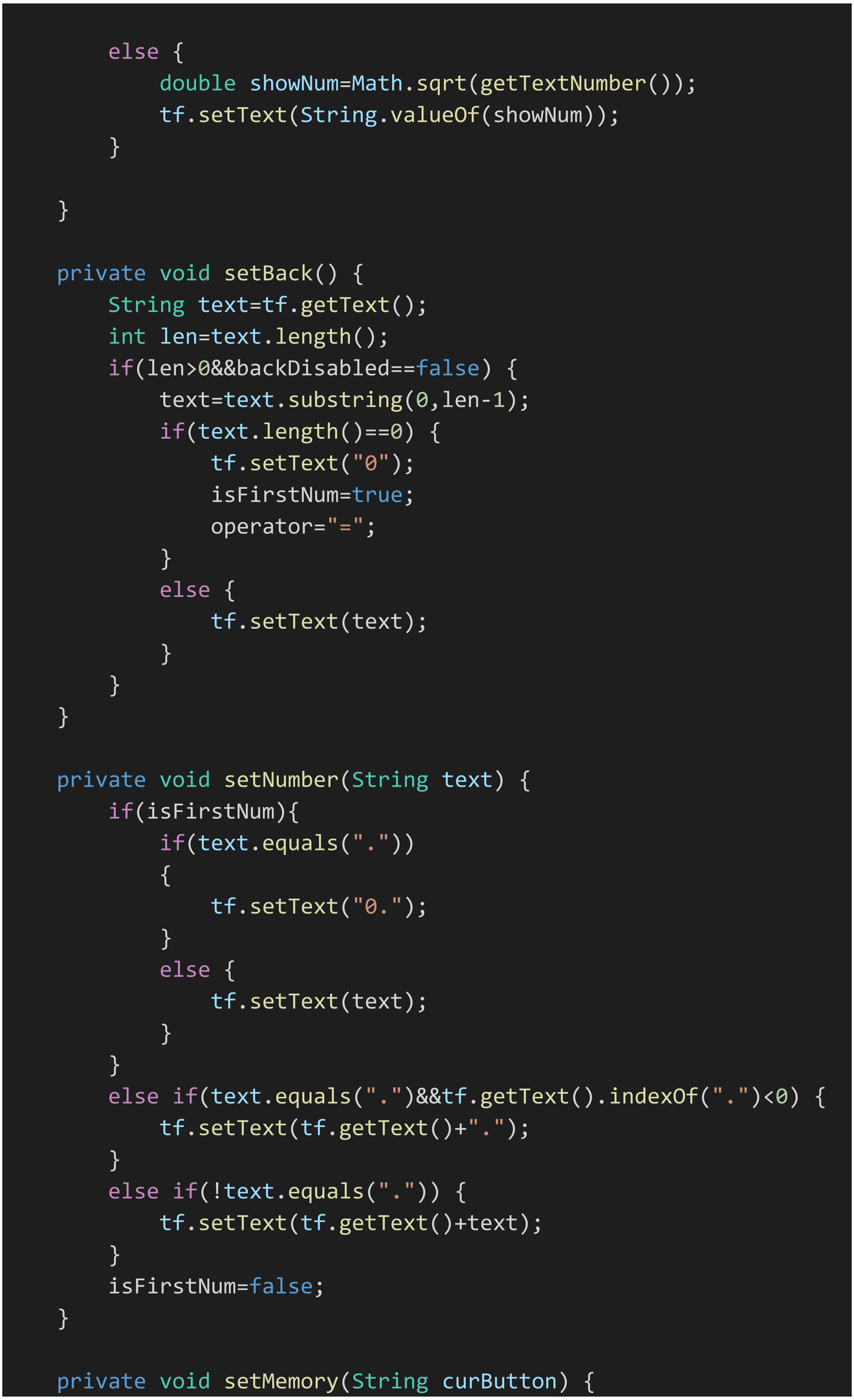
```
            basicButton[i].setBorderPainted(false);
            basicButton[i].setFocusPainted(false);
        }

        JPanel memoryPanel=new JPanel();
        memoryPanel.setLayout(new GridLayout(1,5,3,3));
        memoryPanel.setBackground(BG_ALL);
        Dimension dimensionMemory=new Dimension(200,20);
        memoryPanel.setPreferredSize(dimensionMemory);
        for(int i=0;i<memoryOption.length;i++) {
            memoryButton[i]=new JButton(memoryOption[i]);
            memoryPanel.add(memoryButton[i]);
            memoryButton[i].setBackground(BG_ALL);
            memoryButton[i].setForeground(Color.black);
            memoryButton[i].setFont(FONT_MEMORY);
            memoryButton[i].setBorderPainted(false);
            memoryButton[i].setFocusPainted(false);
        }
        memoryButton[0].setEnabled(false);//MC
        memoryButton[1].setEnabled(false);//MR
        //start with no memory storage

        //Panel layout
        JPanel panelLeft=new JPanel();
        panelLeft.setLayout(new BorderLayout(3,3));
        panelLeft.setBackground(BG_ALL);
        panelLeft.add(commandPanel,BorderLayout.NORTH);
        panelLeft.add(mainPanel,BorderLayout.CENTER);
        JPanel textPanel=new JPanel();
        textPanel.setLayout(new BorderLayout());
        textPanel.add(tf,BorderLayout.CENTER);
        textPanel.add(memoryPanel,BorderLayout.SOUTH);
        Dimension dimensionText=new Dimension(333,90);
        textPanel.setPreferredSize(dimensionText);
        getContentPane().setBackground(BG_ALL);
        getContentPane().setLayout(new BorderLayout(3,5));
        getContentPane().add(textPanel,BorderLayout.NORTH);
        getContentPane().add(panelLeft,BorderLayout.CENTER);
        getContentPane().add(basicPanel,BorderLayout.EAST);
        for(int i=0;i<mainOption.length;i++) {
            mainButton[i].addActionListener(this);
        }
        for(int i=0;i<commandOption.length;i++) {
            commandButton[i].addActionListener(this);
```



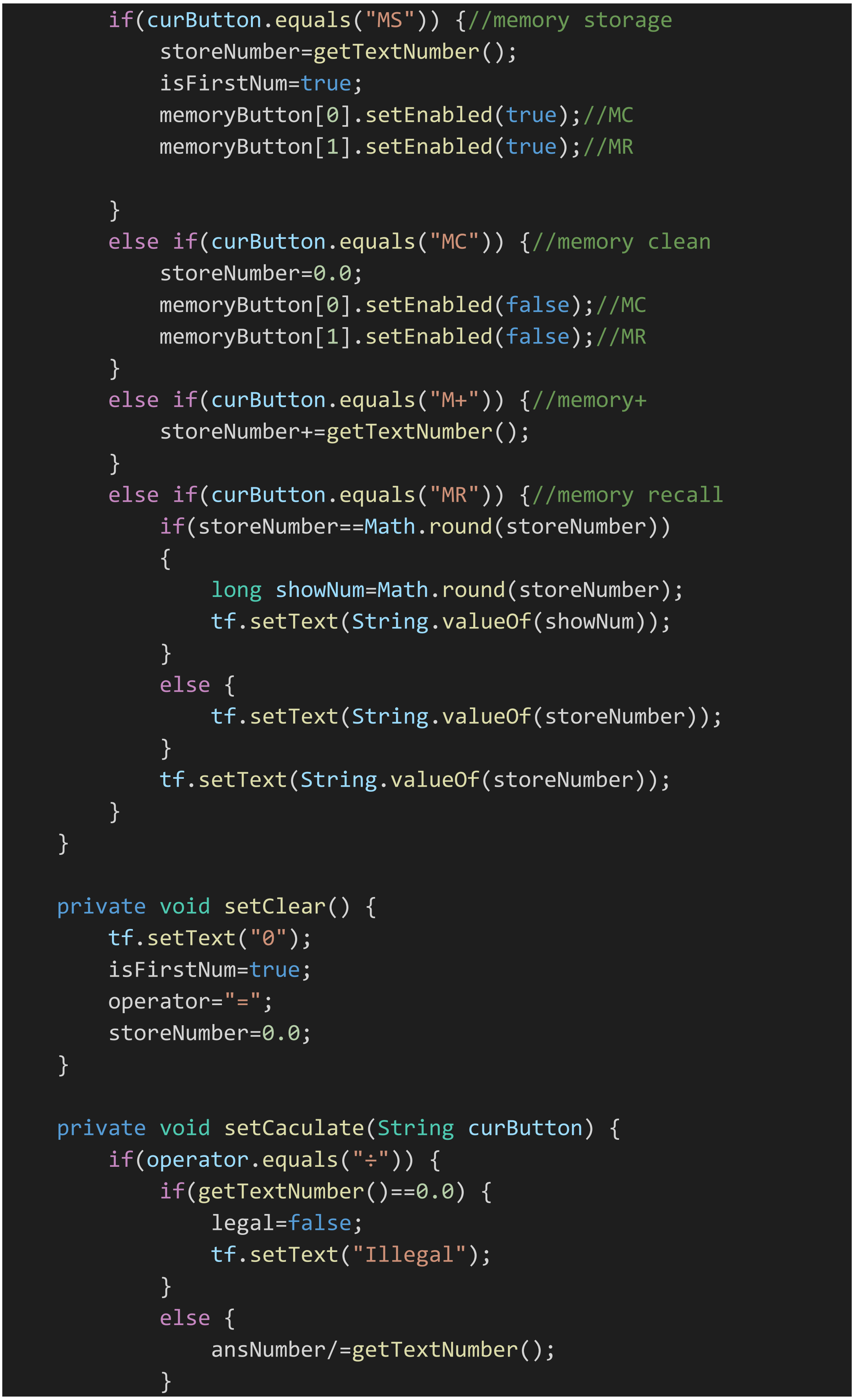
```
        }
        for(int i=0;i<basicOption.length;i++) {
            basicButton[i].addActionListener(this);
        }
        for(int i=0;i<memoryOption.length;i++) {
            memoryButton[i].addActionListener(this);
        }
    }

    //action handler
    public void actionPerformed(ActionEvent e) {
        String curButton=e.getActionCommand();
        if(curButton.equals("")==false) {
            backDisabled=false;
        }
        for(int i=0;i<commandOption.length;i++) {
            if(curButton.equals(commandOption[i]))
            {
                setCommand(curButton);
            }
        }
        //"+/-" is included in command symbol
        if(curButton.equals("+/-")) {
            setCommand(curButton);
        }
        else if(curButton.equals("")) {
            setBack();
        }
        else if("0123456789.".indexOf(curButton)>=0) {
            setNumber(curButton);
        }
        else if(curButton.indexOf("M")!=-1) {
            setMemory(curButton);
        }
        else {
            setCaculate(curButton);
        }
    }

    private void setCommand(String cmd) {
        if(cmd.equals("CE")) {
            tf.setText("0");
          //clean current text
        }
```



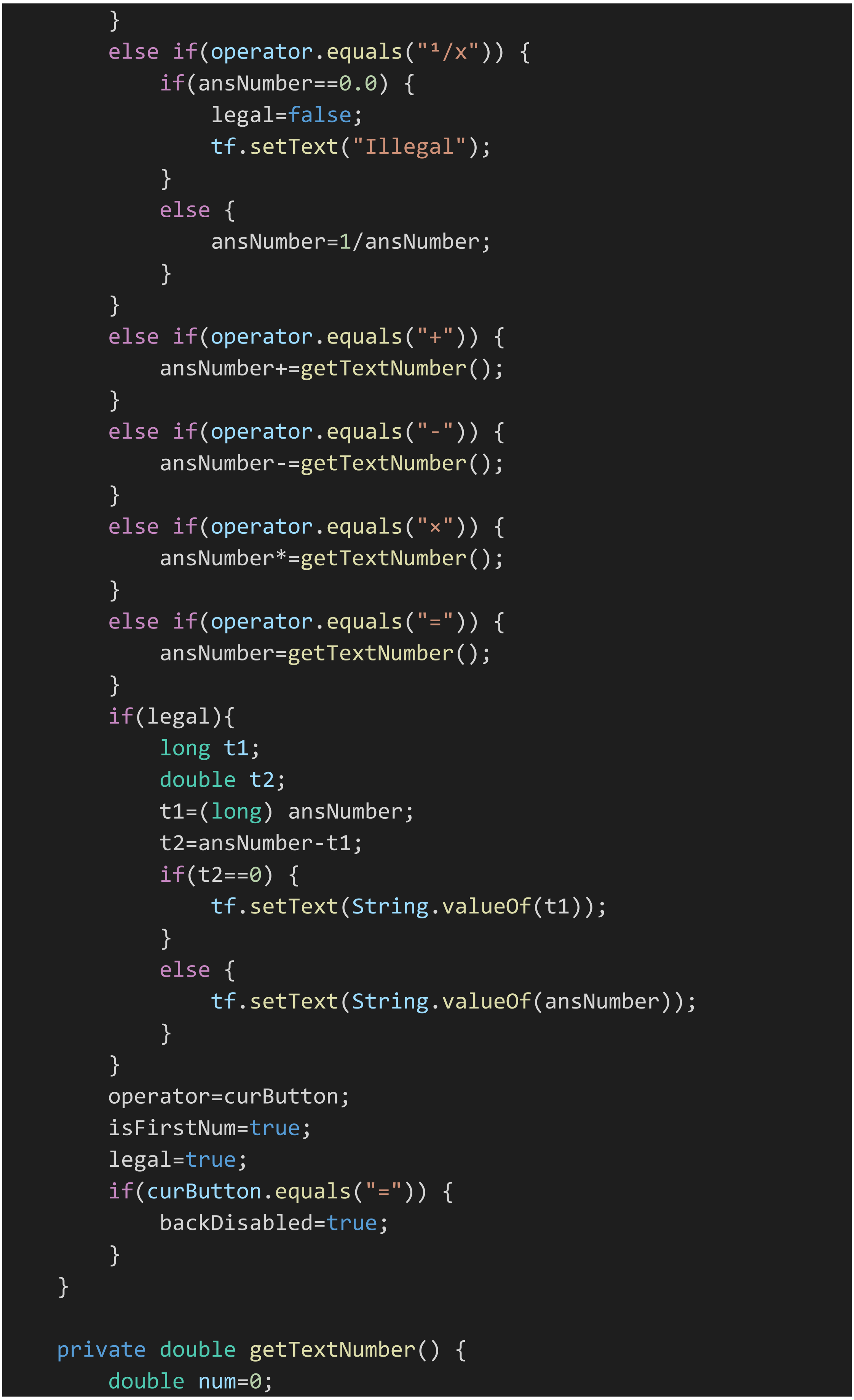
```
        else if(cmd.equals("C")) {
              setClear();
              //clean all
        }
        else if(cmd.equals("%")) {
            double showNum=getTextNumber()*0.01;
            tf.setText(String.valueOf(showNum));
        }
        else if(cmd.equals("¹/x")) {
            if(getTextNumber()==0.0) {
                legal=false;
                tf.setText("Illegal");
            }
            else {
                double showNum=1.0/getTextNumber();
                tf.setText(String.valueOf(showNum));
            }

        }
        else if(cmd.equals("x²")) {
            if(getTextNumber()==Math.round(getTextNumber()))
            {
                long
showNum_long=Math.round(getTextNumber())*Math.round(getTextNumb
er());
                tf.setText(String.valueOf(showNum_long));
            }
            else {
                double showNum=getTextNumber()*getTextNumber();
                tf.setText(String.valueOf(showNum));
            }
        }
        else if(cmd.equals("+/-")) {
            if(getTextNumber()==Math.round(getTextNumber()))
            {
                long showNum_long=-Math.round(getTextNumber());
                tf.setText(String.valueOf(showNum_long));
            }
            else {
                double showNum=-getTextNumber();
                tf.setText(String.valueOf(showNum));
            }

        }
```



```java
        else {
            double showNum=Math.sqrt(getTextNumber());
            tf.setText(String.valueOf(showNum));
        }

    }

    private void setBack() {
        String text=tf.getText();
        int len=text.length();
        if(len>0&&backDisabled==false) {
            text=text.substring(0,len-1);
            if(text.length()==0) {
                tf.setText("0");
                isFirstNum=true;
                operator="=";
            }
            else {
                tf.setText(text);
            }
        }
    }

    private void setNumber(String text) {
        if(isFirstNum){
            if(text.equals("."))
            {
                tf.setText("0.");
            }
            else {
                tf.setText(text);
            }
        }
        else if(text.equals(".")&&tf.getText().indexOf(".")<0) {
            tf.setText(tf.getText()+".");
        }
        else if(!text.equals(".")) {
            tf.setText(tf.getText()+text);
        }
        isFirstNum=false;
    }

    private void setMemory(String curButton) {
```



```
        if(curButton.equals("MS")) {//memory storage
            storeNumber=getTextNumber();
            isFirstNum=true;
            memoryButton[0].setEnabled(true);//MC
            memoryButton[1].setEnabled(true);//MR

        }
        else if(curButton.equals("MC")) {//memory clean
            storeNumber=0.0;
            memoryButton[0].setEnabled(false);//MC
            memoryButton[1].setEnabled(false);//MR
        }
        else if(curButton.equals("M+")) {//memory+
            storeNumber+=getTextNumber();
        }
        else if(curButton.equals("MR")) {//memory recall
            if(storeNumber==Math.round(storeNumber))
            {
                long showNum=Math.round(storeNumber);
                tf.setText(String.valueOf(showNum));
            }
            else {
                tf.setText(String.valueOf(storeNumber));
            }
            tf.setText(String.valueOf(storeNumber));
        }
    }

    private void setClear() {
        tf.setText("0");
        isFirstNum=true;
        operator="=";
        storeNumber=0.0;
    }

    private void setCaculate(String curButton) {
        if(operator.equals("÷")) {
            if(getTextNumber()==0.0) {
                legal=false;
                tf.setText("Illegal");
            }
            else {
                ansNumber/=getTextNumber();
            }
```



```
        }
        else if(operator.equals("¹/x")) {
            if(ansNumber==0.0) {
                legal=false;
                tf.setText("Illegal");
            }
            else {
                ansNumber=1/ansNumber;
            }
        }
        else if(operator.equals("+")) {
            ansNumber+=getTextNumber();
        }
        else if(operator.equals("-")) {
            ansNumber-=getTextNumber();
        }
        else if(operator.equals("×")) {
            ansNumber*=getTextNumber();
        }
        else if(operator.equals("=")) {
            ansNumber=getTextNumber();
        }
        if(legal){
            long t1;
            double t2;
            t1=(long) ansNumber;
            t2=ansNumber-t1;
            if(t2==0) {
                tf.setText(String.valueOf(t1));
            }
            else {
                tf.setText(String.valueOf(ansNumber));
            }
        }
        operator=curButton;
        isFirstNum=true;
        legal=true;
        if(curButton.equals("=")) {
            backDisabled=true;
        }
    }

    private double getTextNumber() {
        double num=0;
```

```
      try {
          num=Double.valueOf(tf.getText()).doubleValue();
      }
      catch (NumberFormatException e) {
      }
      return num;
  }
}
```

五、素材来源:

Win10 窗口颜色使用 Pipette 拾取，“back.png” 和 “sqrt.png” 来源于 iconfont。