Programming in JAVA

lecture 5

What is inside source code of the application, class variables

The structure of the GUI source file

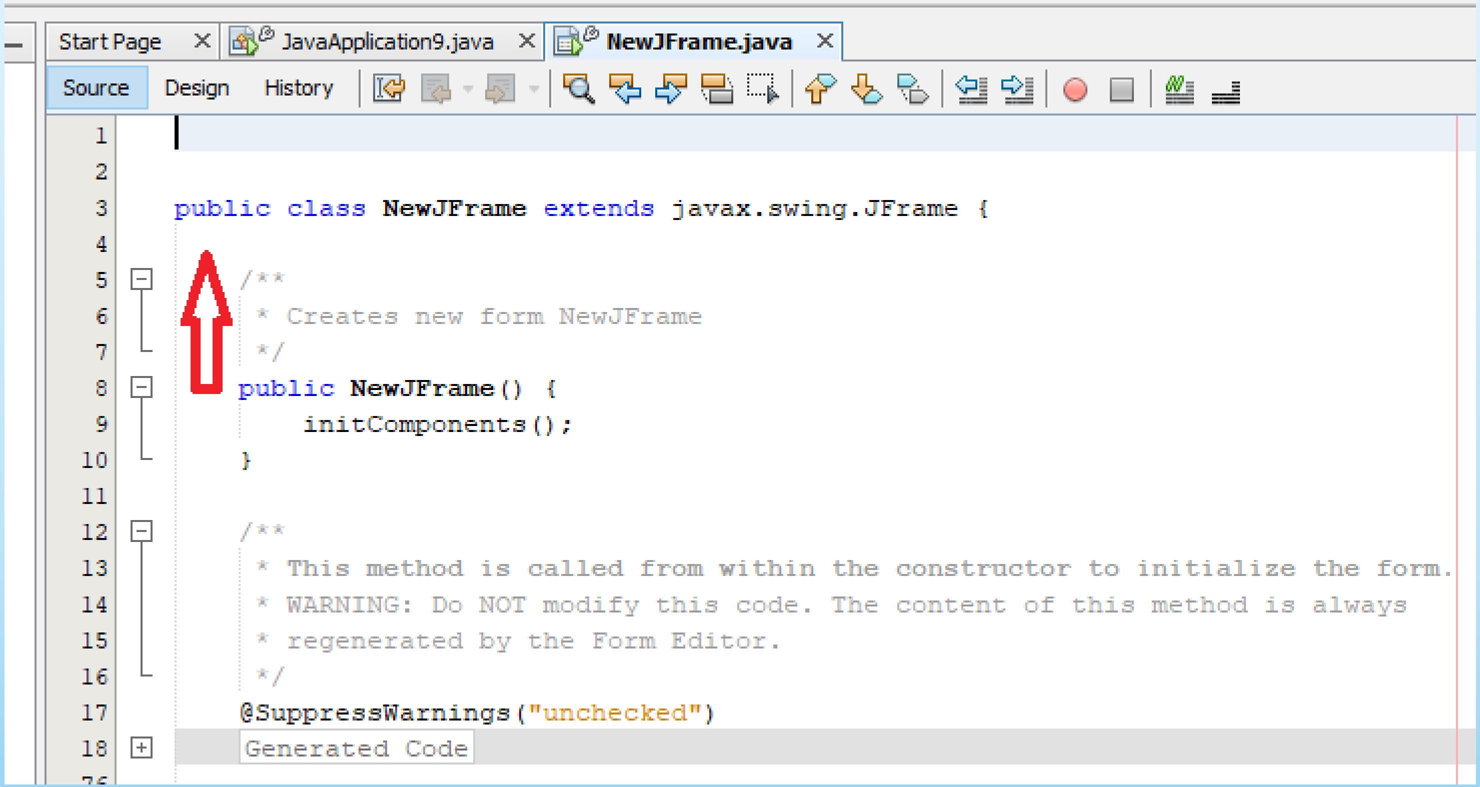
1. Open a GUI project. You can try the example from lecture 4.

2. Switch to source code by pressing source button.

3. Look at the file. On next slide you will get accustomed to diffrent parts of the file.

GUI source file part 1

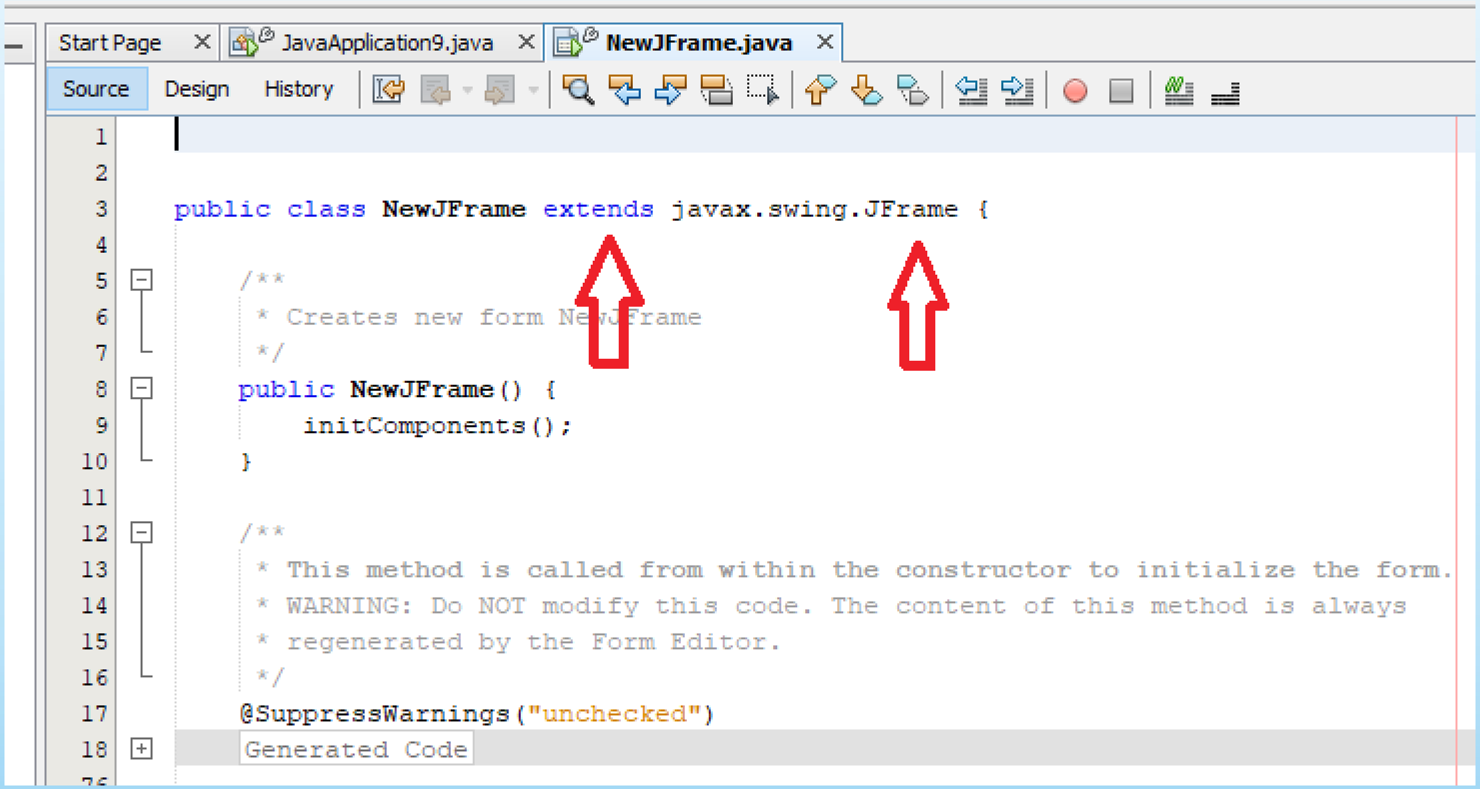
At the begining of the file you can have some import statements.
But the class begins where are the keywords `public class`,



```

public class NewJFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```

GUI source file part 2

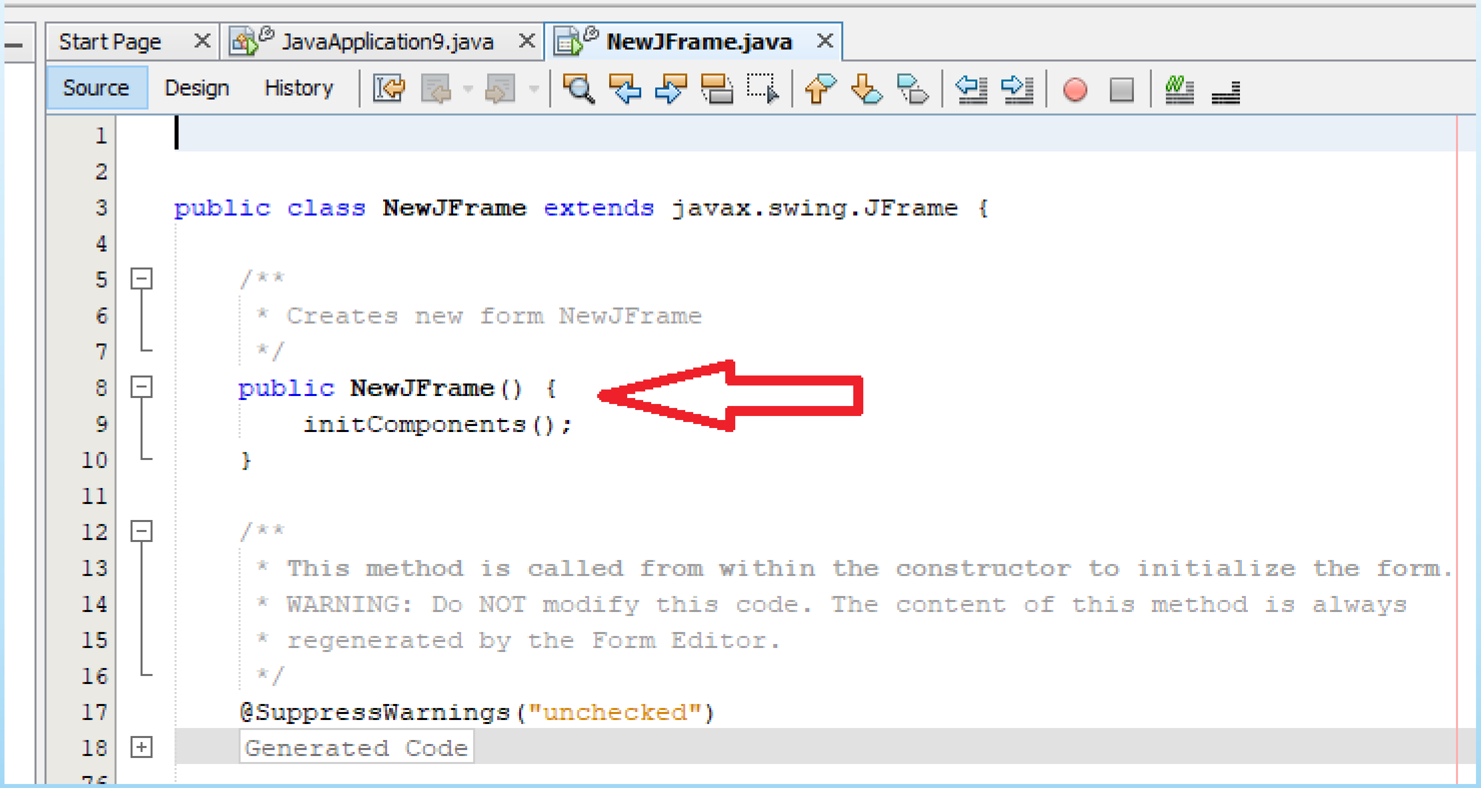
As you can see the class which represents our main window is derived from the JFrame class (that's the meaning of the `extends` keyword). We will talk more about that in the next lecture.



```java

public class NewJFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```

GUI source file part 3

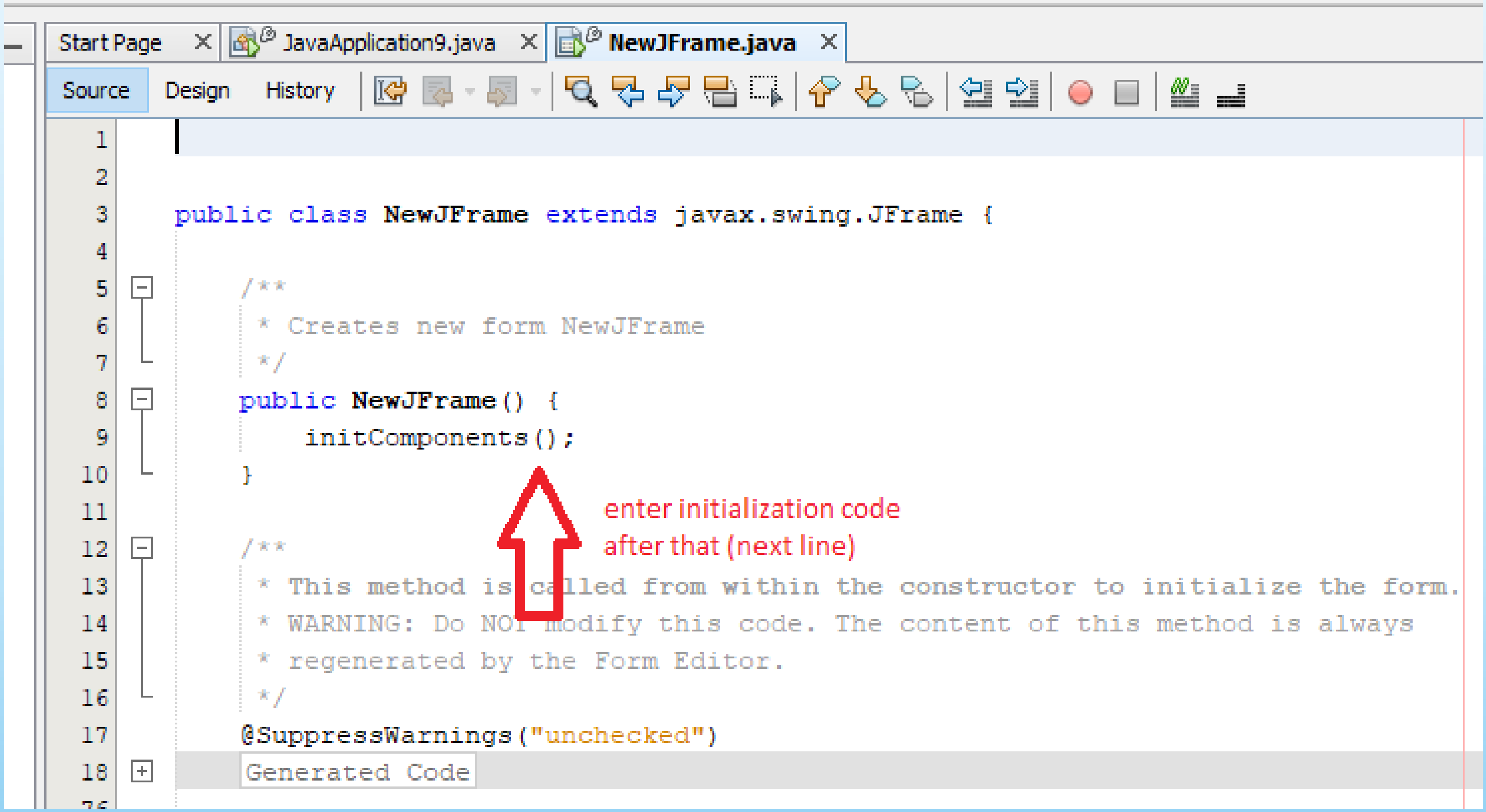
Below you see the constructor of the class. This is a special function that is executed when the object of our class is created. The constructor has the same name as the class and do not have return value type before its name. We will talk more about contructors in the next lecture.



```java

public class NewJFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```

GUI source file part 4

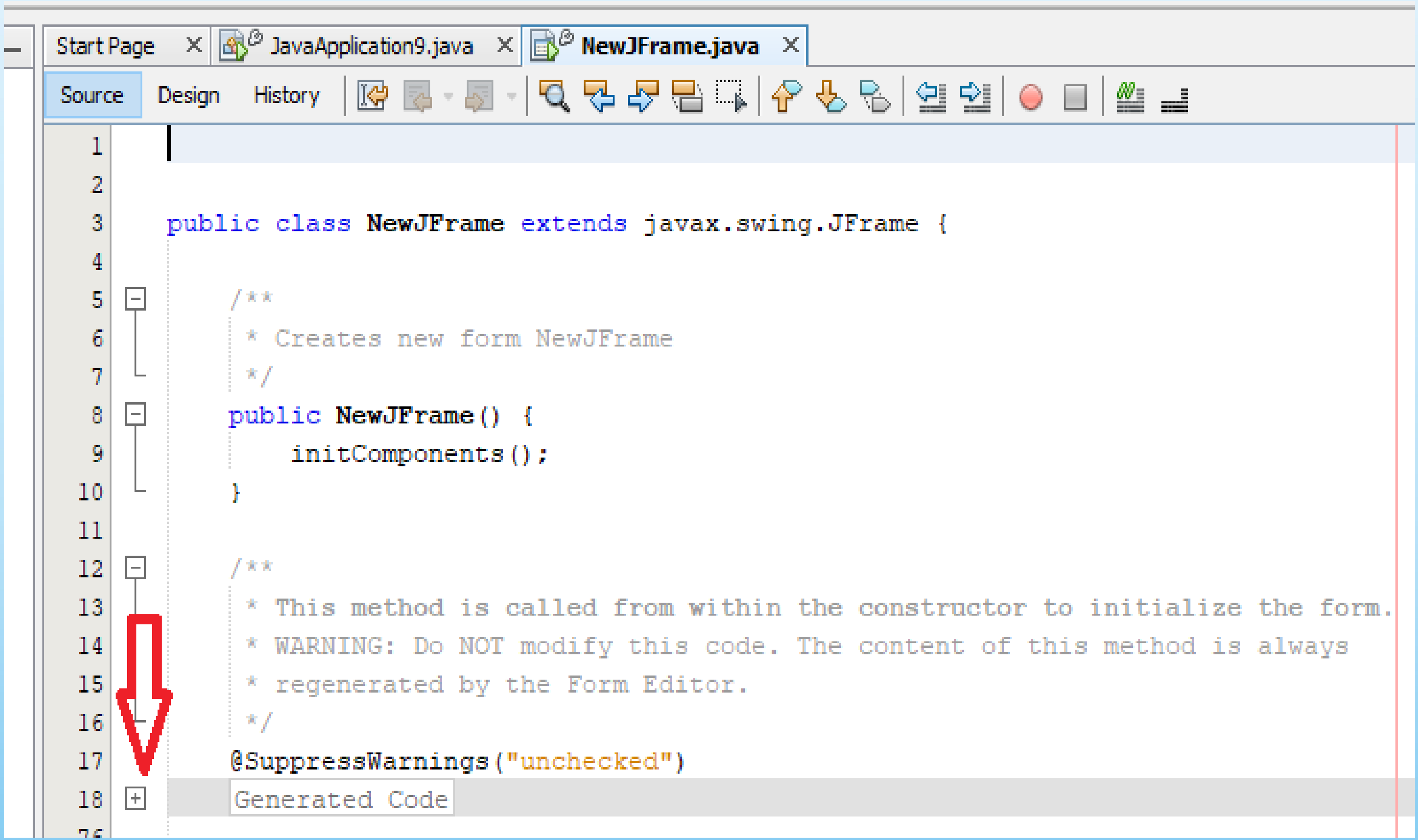
This constructor contains one call to function initComponets.
If you want to do some initialization at start of the application, enter code after the call for function initComponents.
You will need this in exercise 5.



```java

public class NewJFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```

GUI source file part 5

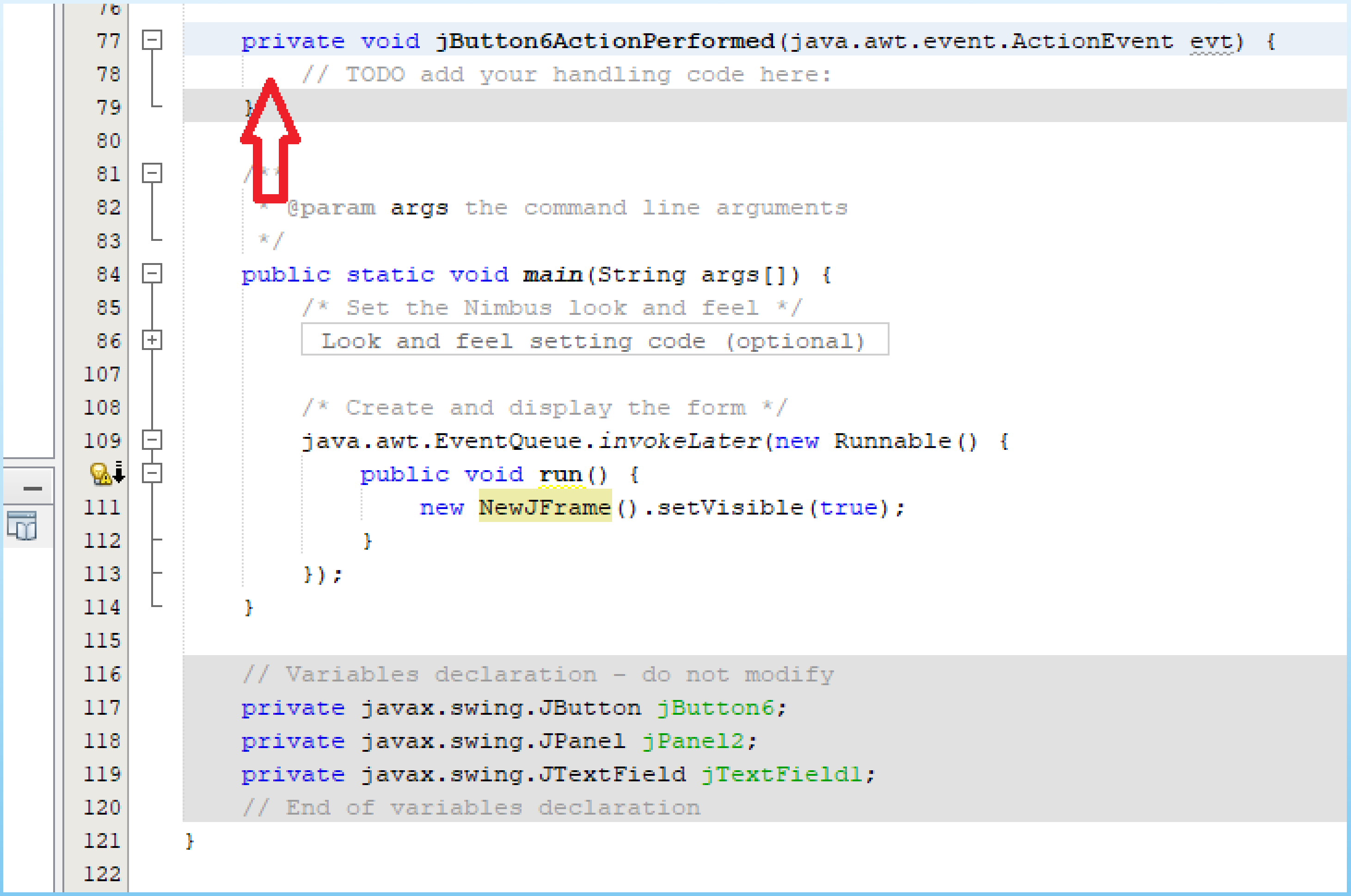
The section Genereted Code contains hidden code that is automatically generated by Netbeans. You are not allowed to change this code. However you can see it if you expand the tree by pressing the + mark. It contains the definition of initComponets function. It sets the configuration of the form layout and binds events into event processing functions.



```
public class NewJFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```

GUI source file part 6

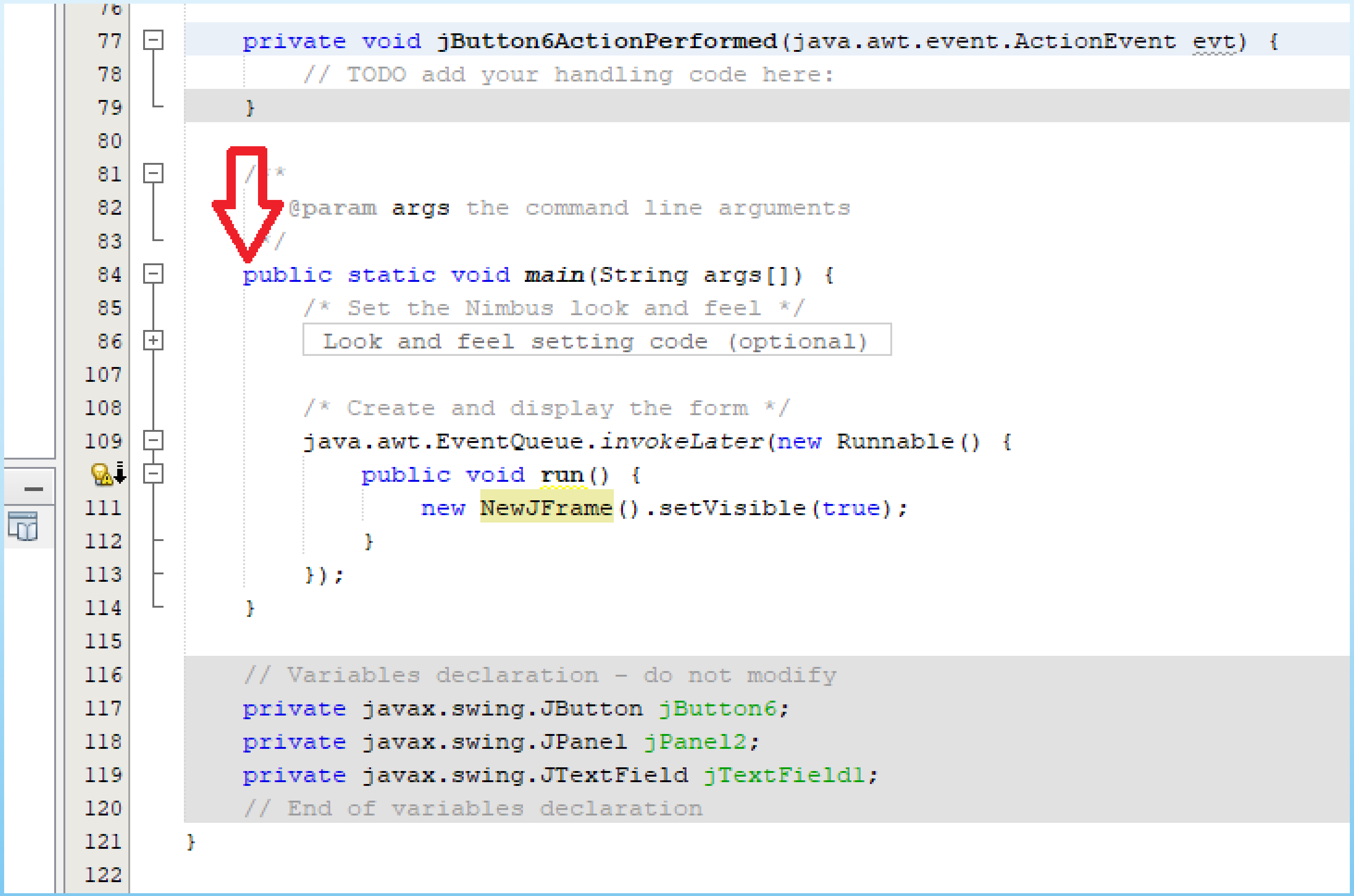
If you added a button, you can have an event handling function inside the file. You put your code here. The function with your code is executed every time the user clicks the button.



```
    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new NewJFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton6;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

GUI source file part 7

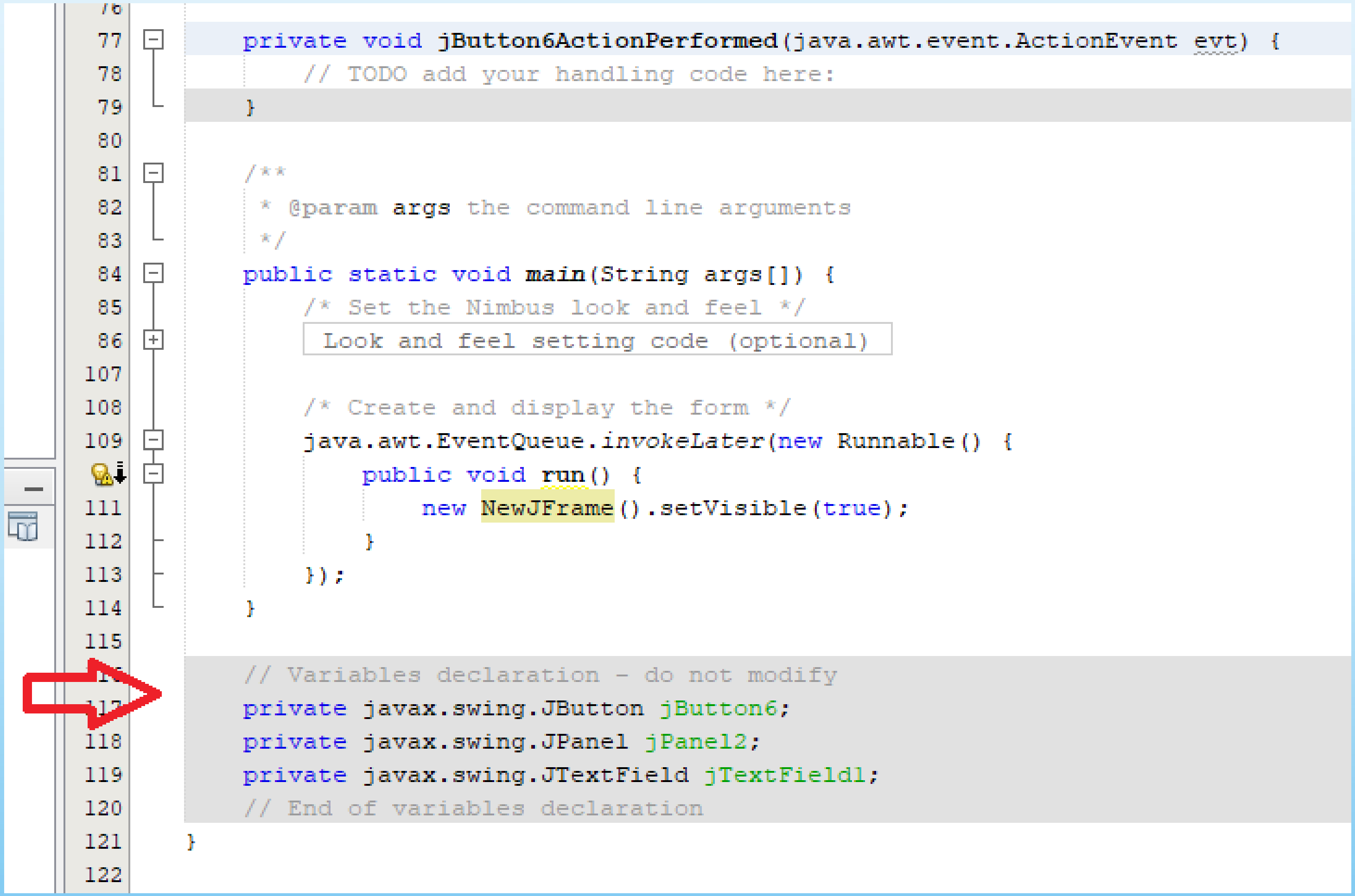
The main function is the entry point for every program. When the operating system starts the application, it sets the instruction pointer to the first operation of the function main. The main function contains the code which creates a new thread for application window. Creating threads is behind the scope of this course because we will not have enough time to deal with it.



```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new NewJFrame().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton6;
private javax.swing.JPanel jPanel2;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}
```

GUI source file part 8

The last thing is the declaration of components inside the frame. Every component that has been added in the design window is here. The code is automatically generated and you cannot change it.



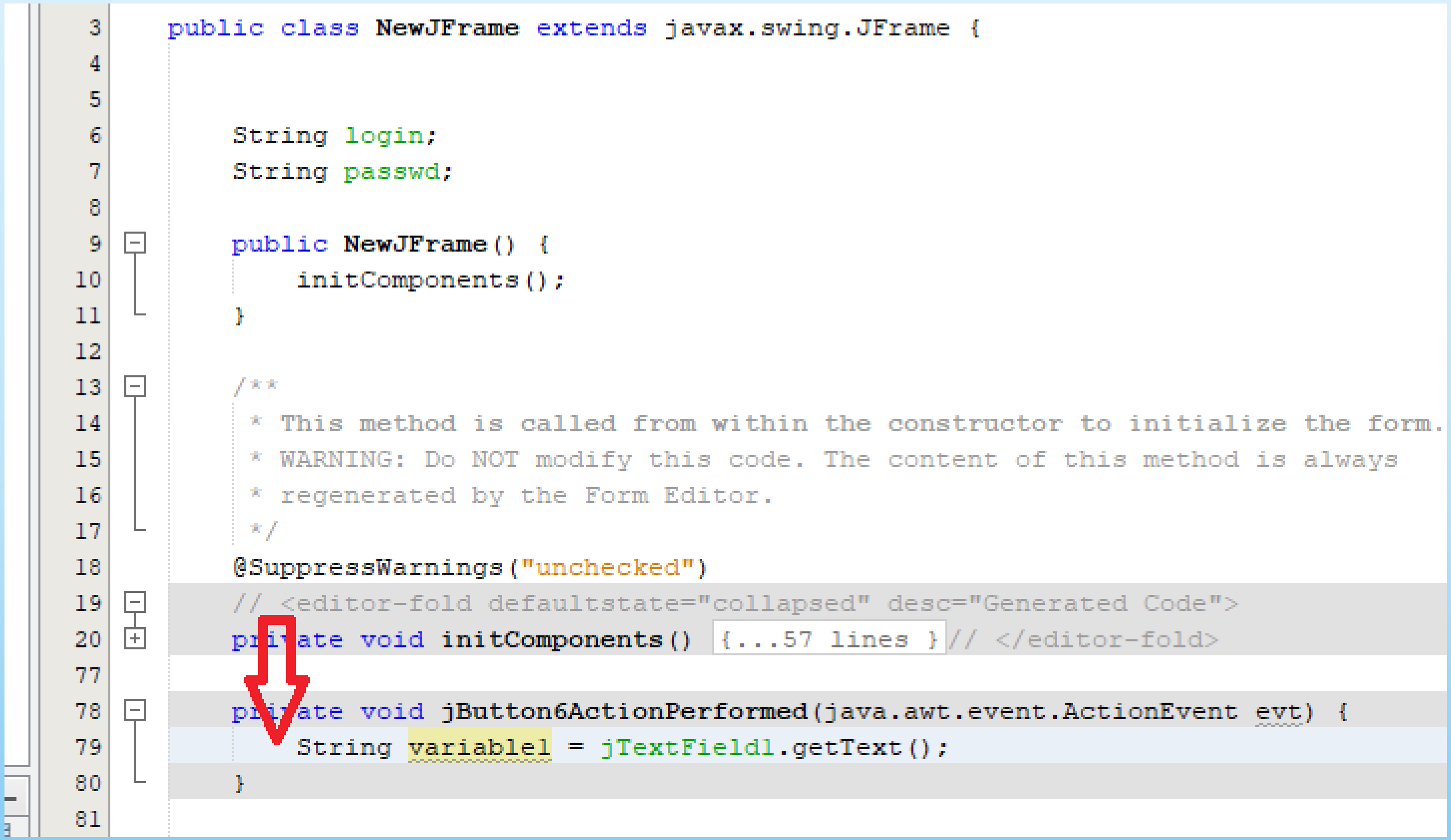
```
    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new NewJFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton6;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

Important remark

needed to do exercise 5

remark

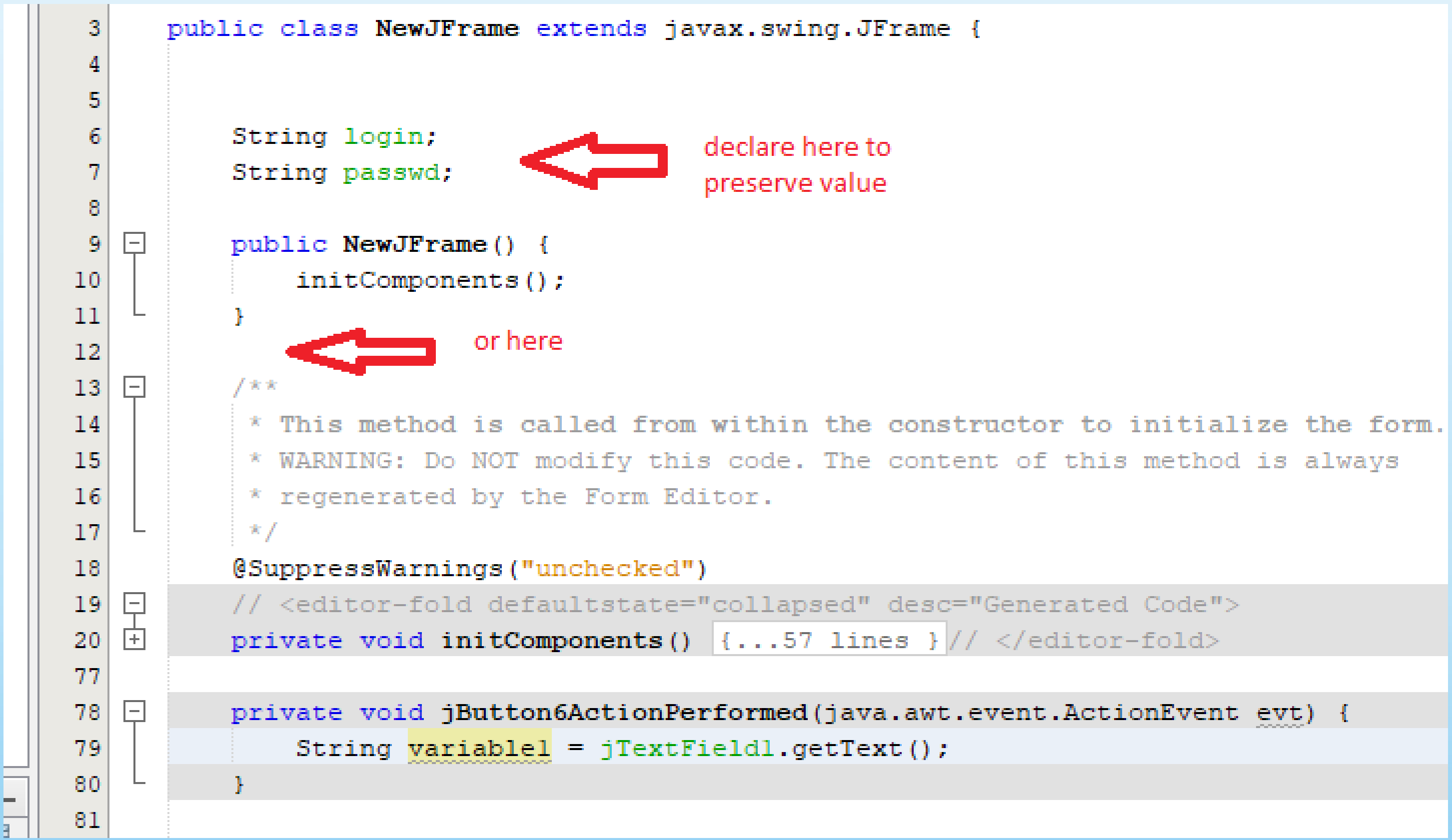
If you declare a variable inside button clicked event handling function, the value of the variable will be destroyed after the function is executed. In the example below the value of variable `variable1` will be lost after function ends.



```
public class NewJFrame extends javax.swing.JFrame {


    String login;
    String passwd;

    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {...57 lines }// </editor-fold>

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        String variable1 = jTextField1.getText();
    }

```

remark

If you want to preserve the value to later use, you have to declare it as a variable of the class (variables `login` and `passwd` in the example below).



```
public class NewJFrame extends javax.swing.JFrame {


    String login;
    String passwd;

    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {...57 lines }// </editor-fold>

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        String variable1 = jTextField1.getText();
    }

```