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.

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

```
Start Page X 30 JavaApplication9.java X 30 NewJFrame.java X
                              public class NewJFrame extends javax.swing.JFrame {
  5
  6
              Creates new form NewJFrame
  8
           public NewJFrame() {
  9
              initComponents();
 10
 11
 12
 13
            * 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
 14
            * regenerated by the Form Editor.
 15
 16
           @SuppressWarnings("unchecked")
 17
           Generated Code
 18
```

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.

```
× JavaApplication9.java × NewJFrame.java ×
               public class NewJFrame extends javax.swing.JFrame {
        public NewJFrame() {
            initComponents();
10
11
12
  13
        * 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
15
         * regenerated by the Form Editor.
16
        @SuppressWarnings("unchecked")
         Generated Code
```

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.

```
Start Page X 3 JavaApplication9.java X 3 NewJFrame.java X
      Design History | 🔀 🖫 - 💹 - | 🔾 🐶 😓 🖫 | 🖟 😓 | 🖆 🖆 | 🥚 🔲 | 🐠 🚅
       public class NewJFrame extends javax.swing.JFrame {
  5
    * Creates new form NewJFrame
           public NewJFrame() {
               initComponents();
 10
 11
 12
            * This method is called from within the constructor to initialize the form.
 13
 14
            * WARNING: Do NOT modify this code. The content of this method is always
 15
            * regenerated by the Form Editor.
            @SuppressWarnings("unchecked")
 17
            Generated Code
```

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.

```
Start Page X 3 JavaApplication9.java X 3 NewJFrame.java X
                          - 🖫 - | 🔍 🐶 😓 🖳 | 🔗 😓 | 🖭 🖭 | 🍥 🗆 | 🕮 🚅
       public class NewJFrame extends javax.swing.JFrame {
  5
             * Creates new form NewJFrame
            public NewJFrame() {
                initComponents();
 10
                                    enter initialization code
 11
                                    after that (next line)
 12
             * This method is called from within the constructor to initialize the form.
 13
             * WARNING: Do NO modify this code. The content of this method is always
 14
             * regenerated by the Form Editor.
 15
 16
            @SuppressWarnings("unchecked")
 17
 18
            Generated Code
```

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.

```
Start Page X Application 9. java X New J Frame. java X
      Design History | 🕝 👼 - 👼 - | 🔩 😓 - 📮 - | 🕰 🚅 - |
       public class NewJFrame extends javax.swing.JFrame {
            * Creates new form NewJFrame
          public NewJFrame() {
               initComponents();
 12
            * This method is called from within the constructor to initialize the form.
 13
            * WARNING: Do NOT modify this code. The content of this method is always
 15
            * regenerated by the Form Editor.
 16
           @SuppressWarnings("unchecked")
 18
           Generated Code
```

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.

```
77
           private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
78
               // TODO add your handling code here:
79
80
    81
82
               param args the command line arguments
83
84
           public static void main(String args[]) {
               /* Set the Nimbus look and feel */
85
86
                Look and feel setting code (optional)
107
108
               /* Create and display the form */
109
               java.awt.EventQueue.invokeLater(new Runnable() {
₩‡
                   public void run() {
111
                       new NewJFrame().setVisible(true);
112
113
               });
114
115
116
           // Variables declaration - do not modify
           private javax.swing.JButton jButton6;
117
118
           private javax.swing.JPanel jPanel2;
           private javax.swing.JTextField jTextFieldl;
119
           // End of variables declaration
120
121
122
```

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.

```
77
           private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
78
               // TODO add your handling code here:
79
80
    _
81
82
               param args the command line arguments
           public static void main(String args[]) {
85
               /* Set the Nimbus look and feel */
    中
                Look and feel setting code (optional)
86
107
               /* Create and display the form */
108
109
               java.awt.EventQueue.invokeLater(new Runnable() {
₩.
                   public void run() {
111
                       new NewJFrame ().setVisible(true);
112
113
               });
114
115
116
           // Variables declaration - do not modify
117
           private javax.swing.JButton jButton6;
           private javax.swing.JPanel jPanel2;
118
119
           private javax.swing.JTextField jTextFieldl;
120
           // End of variables declaration
121
122
```

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.

```
77
           private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
 78
               // TODO add your handling code here:
 79
 82
            * @param args the command line arguments
 83
           public static void main(String args[]) {
               /* Set the Nimbus look and feel */
                Look and feel setting code (optional)
 86
107
               /* Create and display the form */
108
109
               java.awt.EventQueue.invokeLater(new Runnable() {
₩.
                   public void run() {
                       new NewJFrame().setVisible(true);
112
113
               });
114
           // Variables declaration - do not modify
           private javax.swing.JButton jButton6;
           private javax.swing.JPanel jPanel2;
119
           private javax.swing.JTextField jTextFieldl;
120
           // End of variables declaration
121
122
```

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() {
10
              initComponents();
11
12
   13
14
           * 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
15
           * regenerated by the Form Editor.
16
17
          @SuppressWarnings("unchecked")
18
   19
          // <editor-fold defaultstate="collapsed" desc="Generated Code">
          private void initComponents() {...57 lines } // </editor-fold>
20
77
              ate void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
   78
              String variable1 = jTextField1.getText();
79
80
81
```

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;
                                         declare here to
          String passwd;
                                         preserve value
   public NewJFrame() {
10
              initComponents();
11
                          or here
12
13
   14
           * 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
15
16
           * regenerated by the Form Editor.
17
18
          @SuppressWarnings("unchecked")
          // <editor-fold defaultstate="collapsed" desc="Generated Code">
19
   \Box
          private void initComponents() {...57 lines } // </editor-fold>
20
77
78
          private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
79
              String variablel = jTextFieldl.getText();
80
81
```