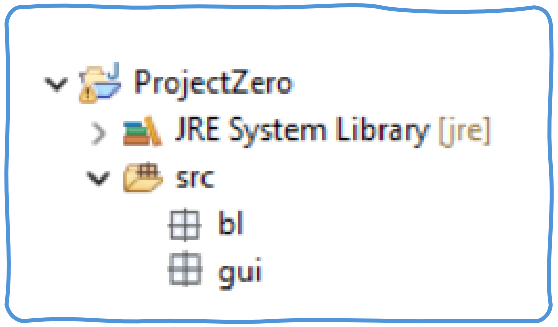
Anwendungsentwicklung

Eine simple Anwendung mit zwei Schichten:

gui	mit javax.swing	Schritte 0-10
Persistence	mit java.io.Serialization	TODO

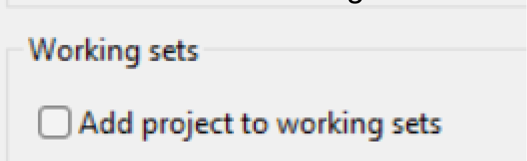
▪▪▪@Dr. V. Tikko

0 In Eclipse: new Project; new package; new package



```
ProjectZero
  JRE System Library [jre]
  src
    bl
    gui
```

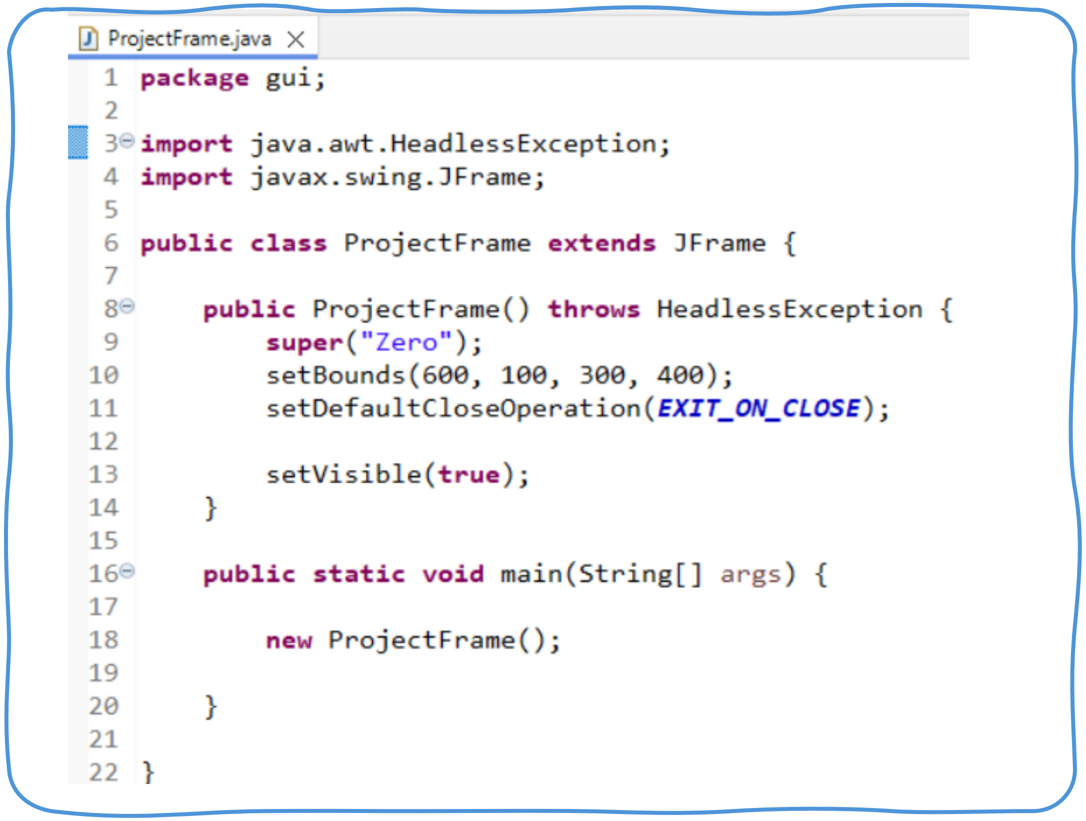
Achtung: Diese Checkbox soll NICHT angeklickt sein!



Working sets

☐ Add project to working sets

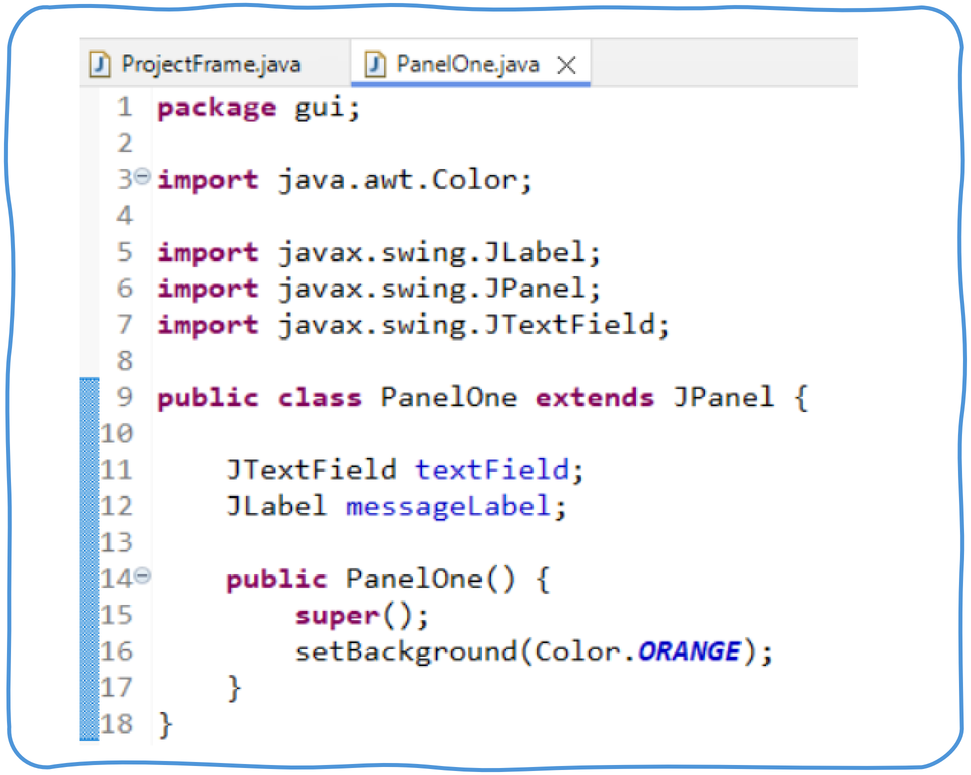
1 Im package gui: new Class ProjectFrame extends Jframe; main-Method



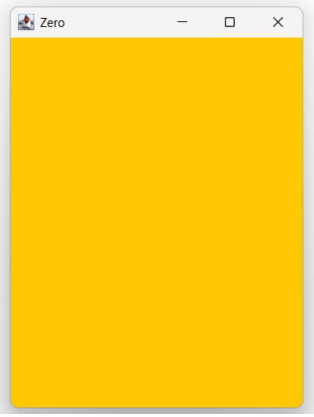
ProjectFrame.java

```java
package gui;

import java.awt.HeadlessException;
import javax.swing.JFrame;

public class ProjectFrame extends JFrame {

    public ProjectFrame() throws HeadlessException {
        super("Zero");
        setBounds(600, 100, 300, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        setVisible(true);
    }

    public static void main(String[] args) {

        new ProjectFrame();

    }

}
```

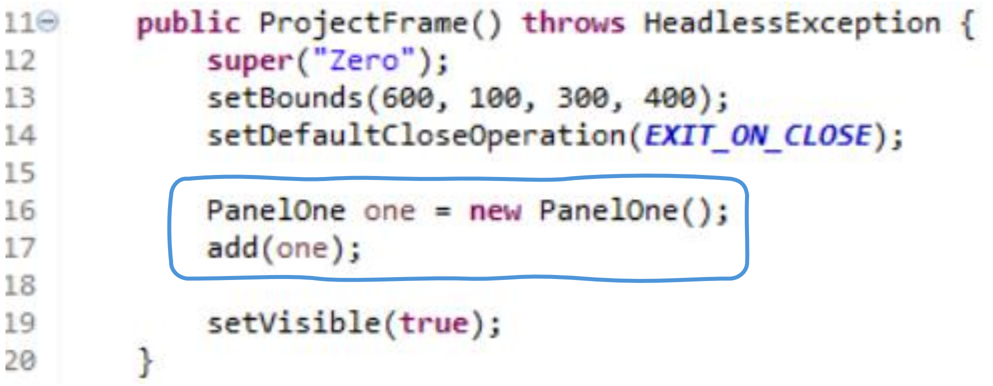
2 In gui: new Class PanelOne extends JPanel



```java
package gui;

import java.awt.Color;

import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class PanelOne extends JPanel {

    JTextField textField;
    JLabel messageLabel;

    public PanelOne() {
        super();
        setBackground(Color.ORANGE);
    }
}
```

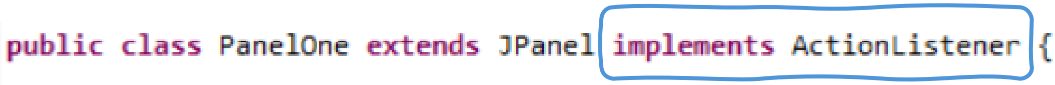


3 Im Konstruktor von ProjectFrame:



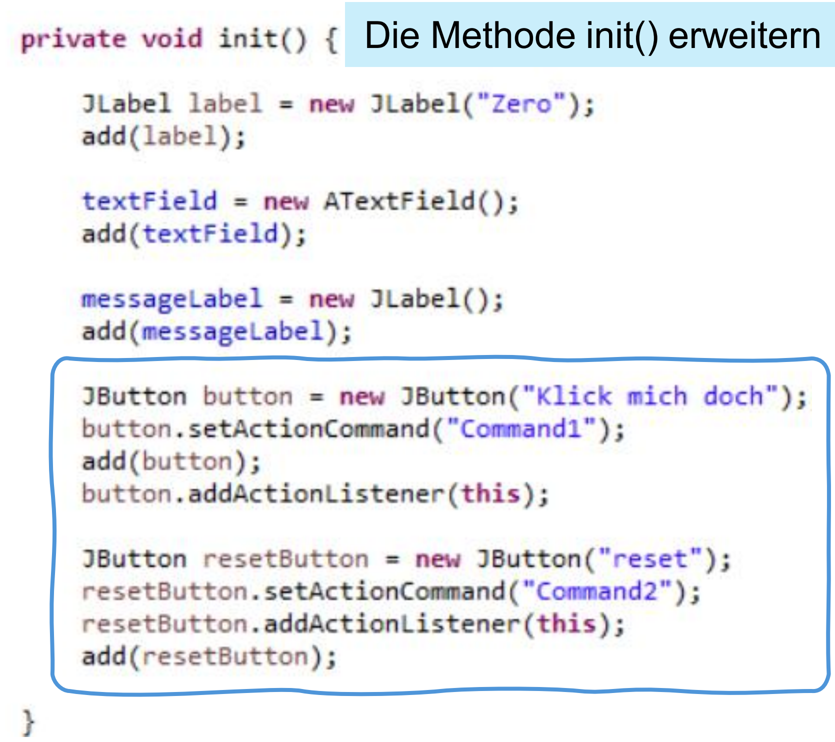
```java
    public ProjectFrame() throws HeadlessException {
        super("Zero");
        setBounds(600, 100, 300, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        PanelOne one = new PanelOne();
        add(one);

        setVisible(true);
    }
```

4 Zwei Buttons werden zu PanelOne hinzugefügt:

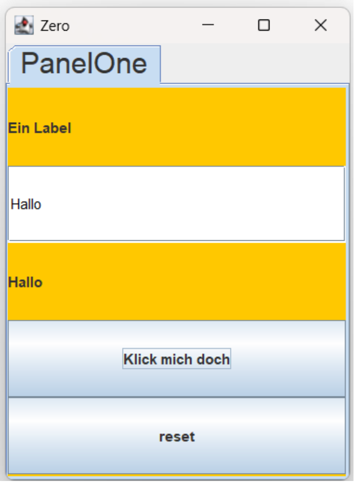


```java
public class PanelOne extends JPanel implements ActionListener {
```

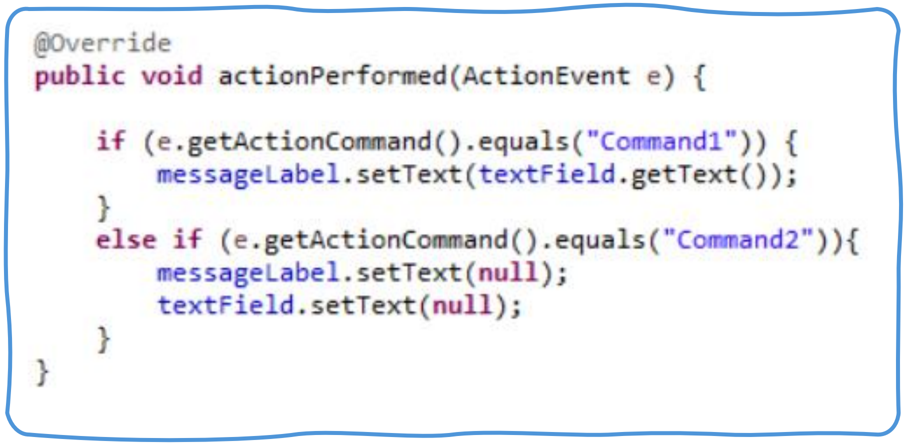
Die Methode init() erweitern



```java
private void init() {

    JLabel label = new JLabel("Zero");
    add(label);

    textField = new ATextField();
    add(textField);

    messageLabel = new JLabel();
    add(messageLabel);

    JButton button = new JButton("Klick mich doch");
    button.setActionCommand("Command1");
    add(button);
    button.addActionListener(this);

    JButton resetButton = new JButton("reset");
    resetButton.setActionCommand("Command2");
    resetButton.addActionListener(this);
    add(resetButton);

}
```



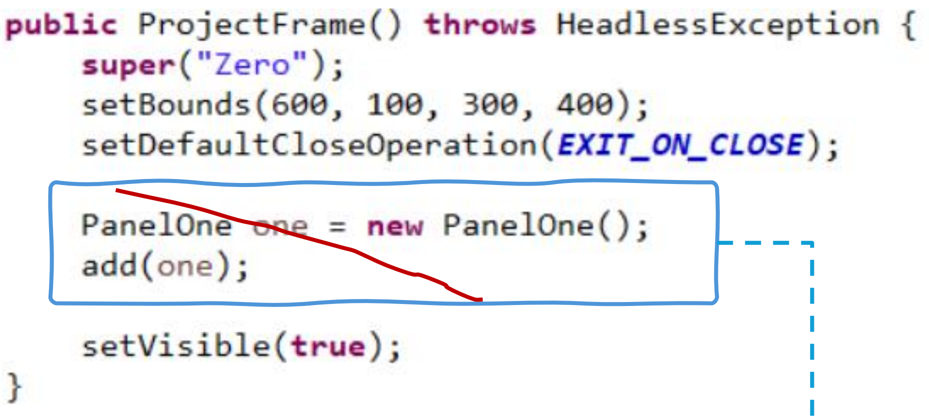
Die Methode actionPerformed implementieren()



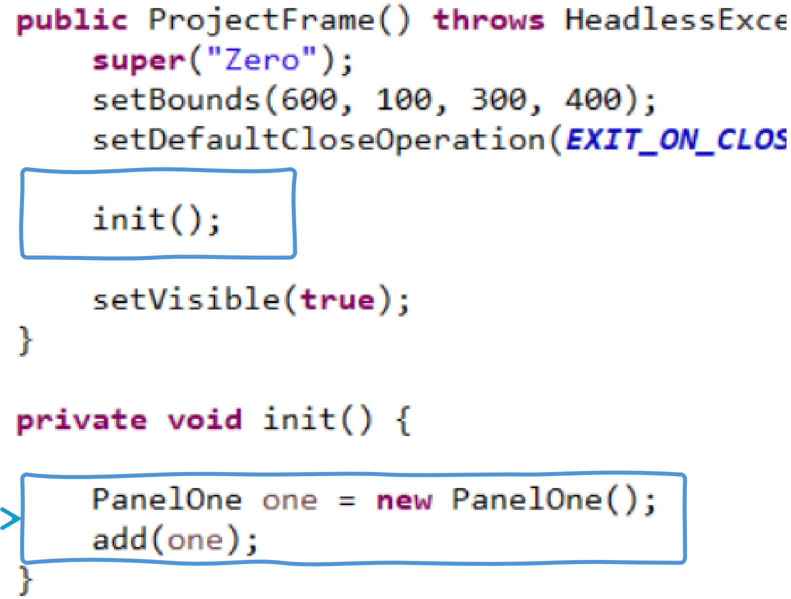
```java
@Override
public void actionPerformed(ActionEvent e) {

    if (e.getActionCommand().equals("Command1")) {
        messageLabel.setText(textField.getText());
    }
    else if (e.getActionCommand().equals("Command2")){
        messageLabel.setText(null);
        textField.setText(null);
    }
}
```

5

In ProjectFrame:

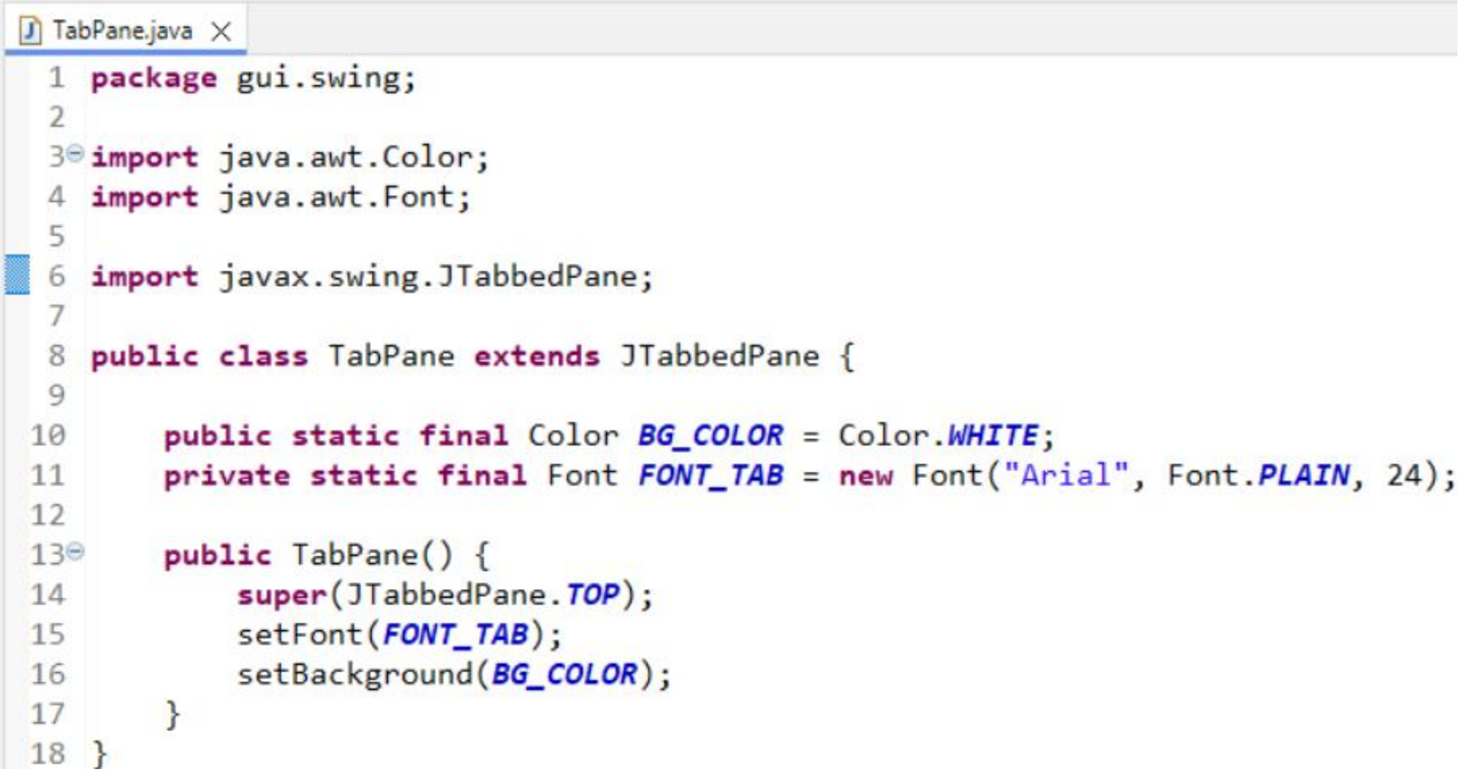


```
public ProjectFrame() throws HeadlessException {
    super("Zero");
    setBounds(600, 100, 300, 400);
    setDefaultCloseOperation(EXIT_ON_CLOSE);

    PanelOne one = new PanelOne();
    add(one);

    setVisible(true);
}
```



```
public ProjectFrame() throws HeadlessExce
    super("Zero");
    setBounds(600, 100, 300, 400);
    setDefaultCloseOperation(EXIT_ON_CLOS

    init();

    setVisible(true);
}

private void init() {

    PanelOne one = new PanelOne();
    add(one);
}
```

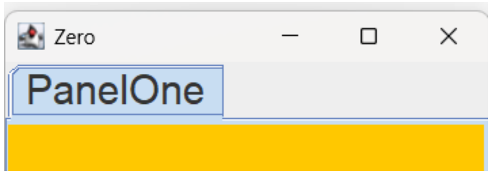
6 In gui: new Class TabPane extends JTabbedPane

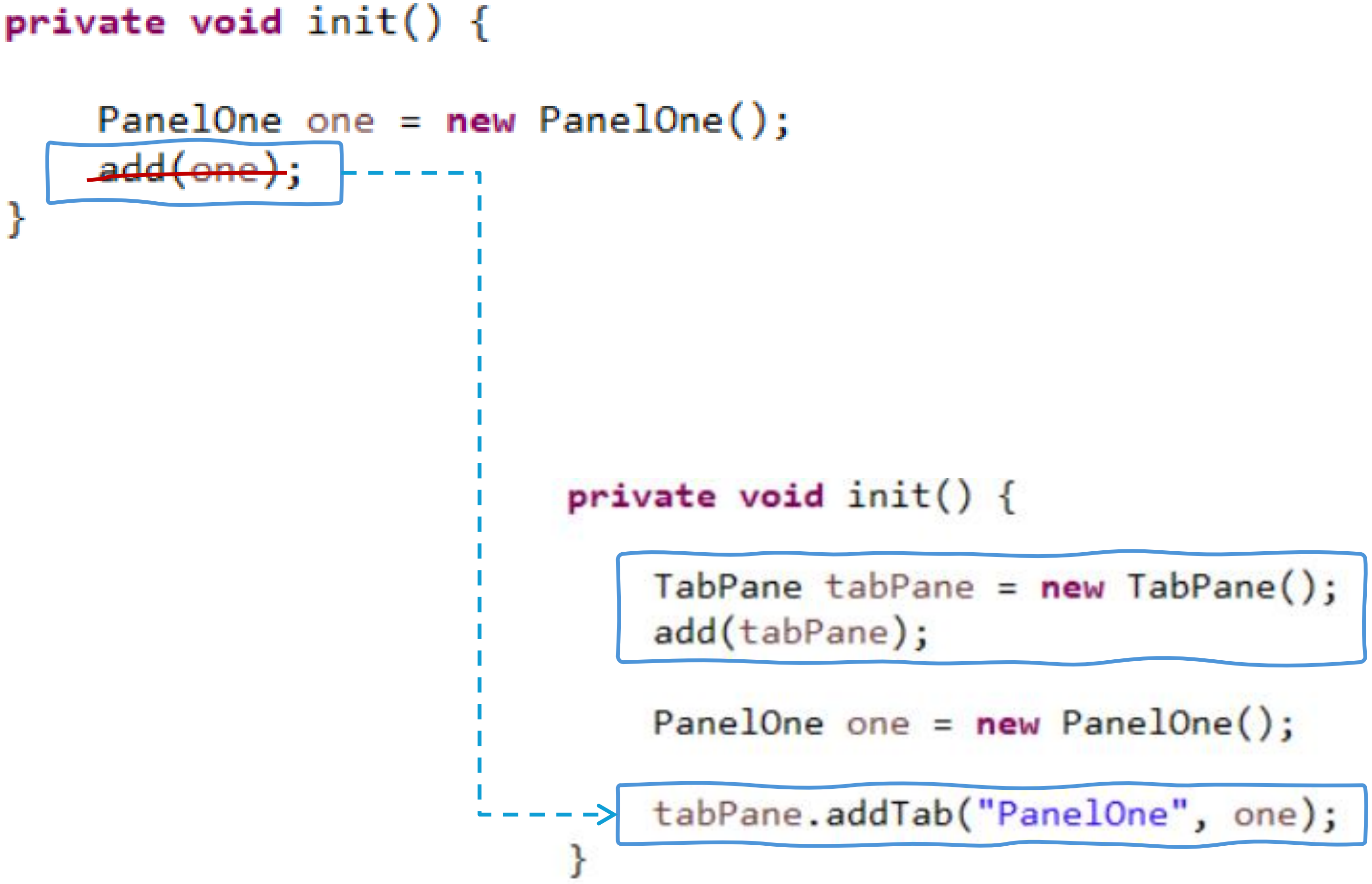


```java
package gui.swing;

import java.awt.Color;
import java.awt.Font;

import javax.swing.JTabbedPane;

public class TabPane extends JTabbedPane {

    public static final Color BG_COLOR = Color.WHITE;
    private static final Font FONT_TAB = new Font("Arial", Font.PLAIN, 24);

    public TabPane() {
        super(JTabbedPane.TOP);
        setFont(FONT_TAB);
        setBackground(BG_COLOR);
    }
}
```

7

In der init-Methode der Klasse ProjectFrame





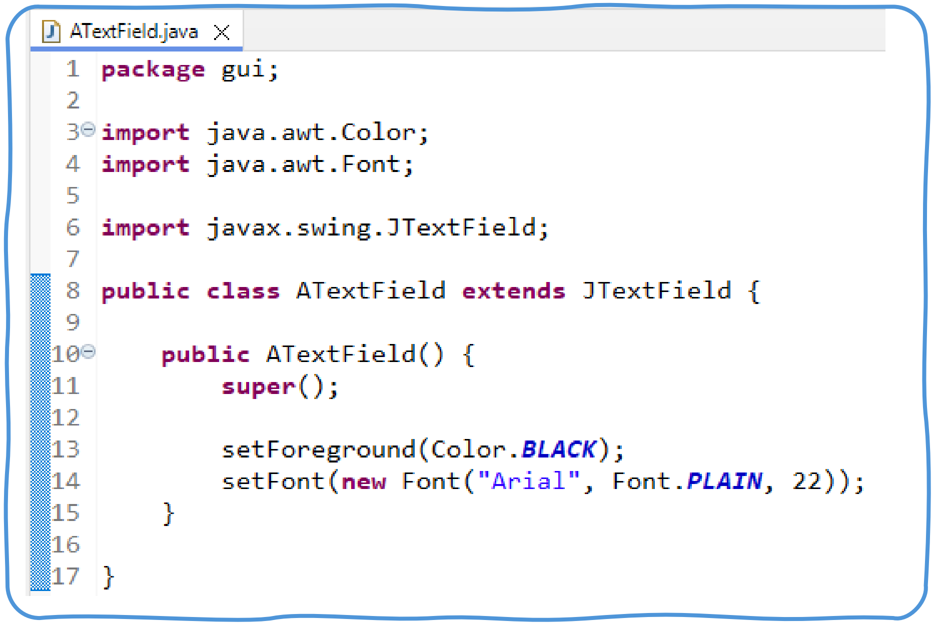
```
private void init() {

    PanelOne one = new PanelOne();
    ~~add(one)~~;
}
```

```
private void init() {

    TabPane tabPane = new TabPane();
    add(tabPane);

    PanelOne one = new PanelOne();

    tabPane.addTab("PanelOne", one);
}
```

8

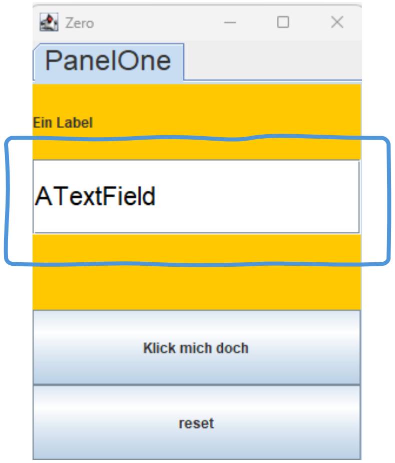
Eigene Klassen für **alle** gui-Elemente schreiben und benutzen

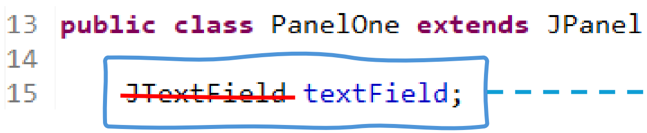
Beispielf für das JTextField



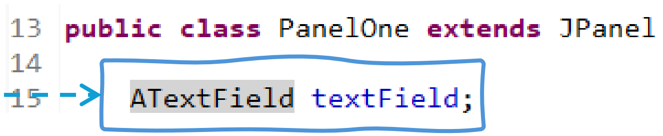
ATextField.java

```java
package gui;

import java.awt.Color;
import java.awt.Font;

import javax.swing.JTextField;

public class ATextField extends JTextField {

    public ATextField() {
        super();

        setForeground(Color.BLACK);
        setFont(new Font("Arial", Font.PLAIN, 22));
    }

}
```

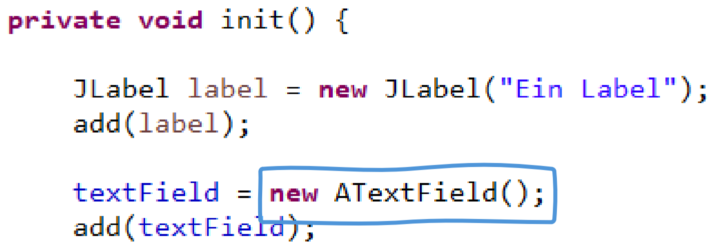




```java
public class PanelOne extends JPanel

    ~~JTextField~~ textField;
```



```java
public class PanelOne extends JPanel

    ATextField textField;
```

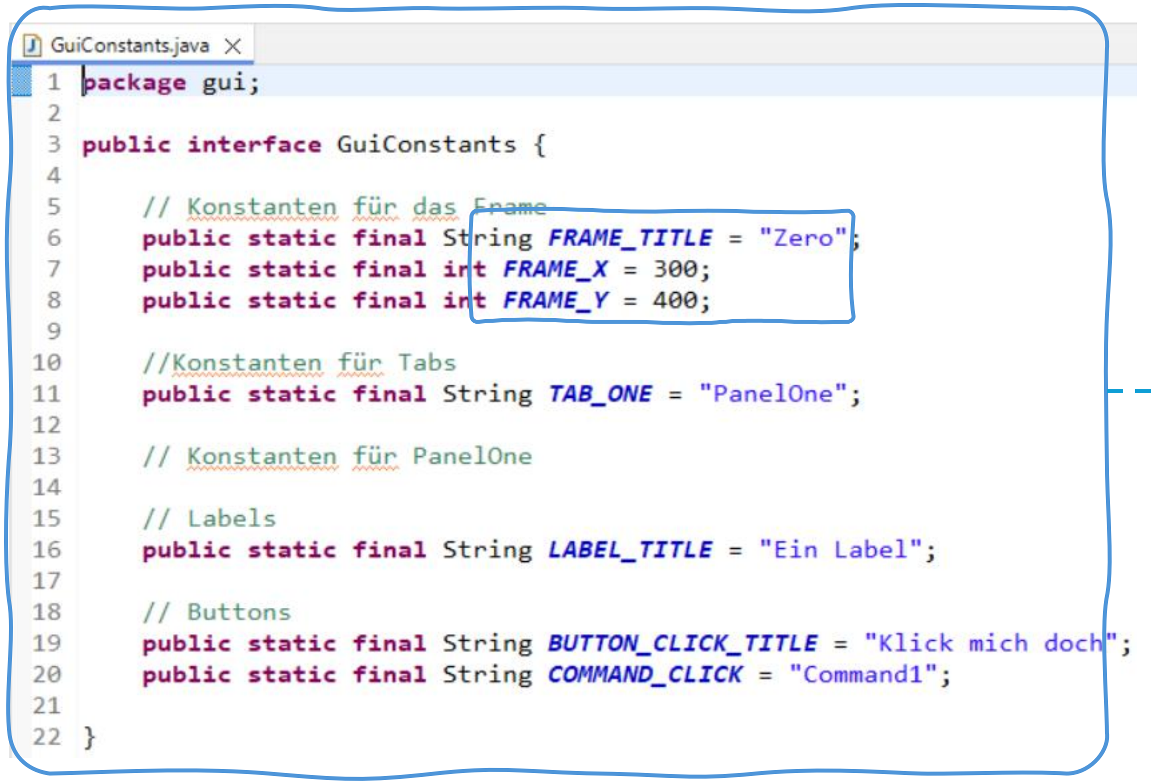


```java
private void init() {

    JLabel label = new JLabel("Ein Label");
    add(label);

    textField = new ATextField();
    add(textField);
```

9

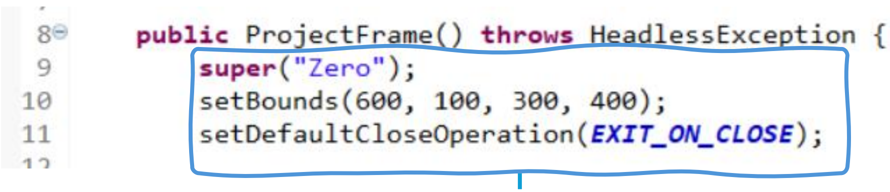
Interface GuiConstants und die Verwendung in ProjectFrame

TODO:
Vervollständigen

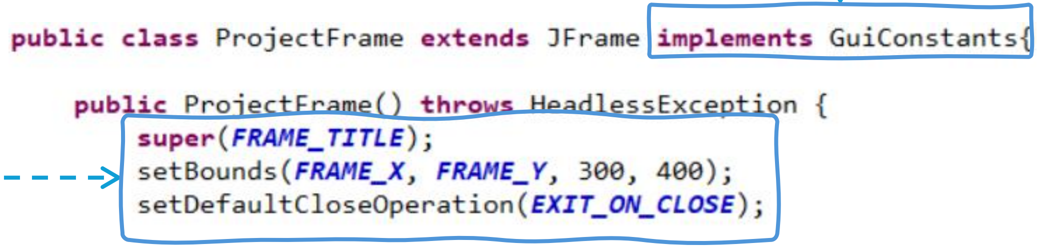


GuiConstants.java

```java
package gui;

public interface GuiConstants {

    // Konstanten für das Frame
    public static final String FRAME_TITLE = "Zero";
    public static final int FRAME_X = 300;
    public static final int FRAME_Y = 400;

    //Konstanten für Tabs
    public static final String TAB_ONE = "PanelOne";

    // Konstanten für PanelOne

    // Labels
    public static final String LABEL_TITLE = "Ein Label";

    // Buttons
    public static final String BUTTON_CLICK_TITLE = "Klick mich doch";
    public static final String COMMAND_CLICK = "Command1";

}
```



```java
    public ProjectFrame() throws HeadlessException {
        super("Zero");
        setBounds(600, 100, 300, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
```

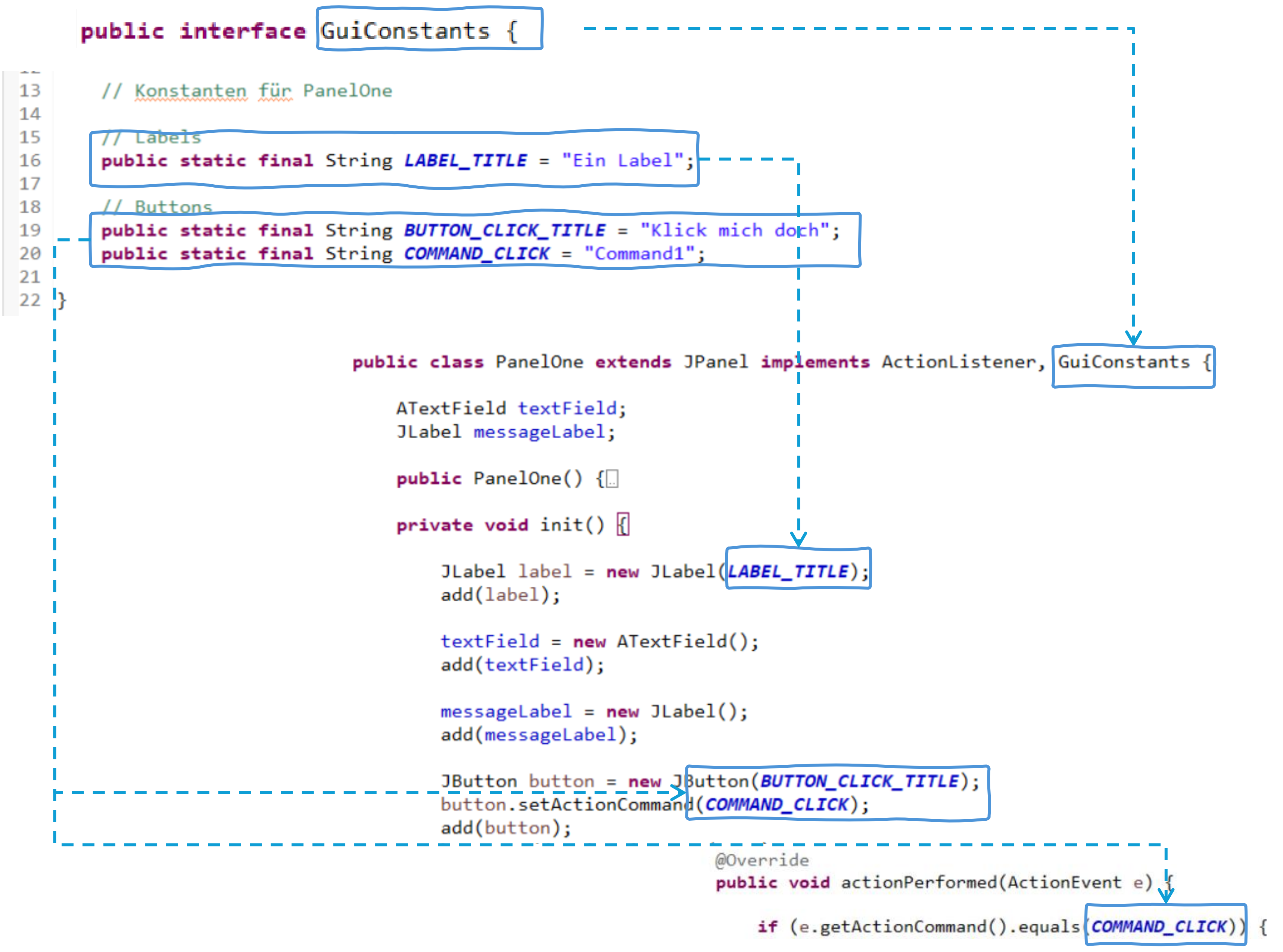


```java
public class ProjectFrame extends JFrame implements GuiConstants{

    public ProjectFrame() throws HeadlessException {
        super(FRAME_TITLE);
        setBounds(FRAME_X, FRAME_Y, 300, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
```

10

Interface GuiConstants und die Verwendung in PanelOne

TODO:
Vervollständigen



```
public interface GuiConstants {

    // Konstanten für PanelOne

    // Labels
    public static final String LABEL_TITLE = "Ein Label";

    // Buttons
    public static final String BUTTON_CLICK_TITLE = "Klick mich doch";
    public static final String COMMAND_CLICK = "Command1";

}
```

```
public class PanelOne extends JPanel implements ActionListener, GuiConstants {

    ATextField textField;
    JLabel messageLabel;

    public PanelOne() {□

    private void init() {

        JLabel label = new JLabel(LABEL_TITLE);
        add(label);

        textField = new ATextField();
        add(textField);

        messageLabel = new JLabel();
        add(messageLabel);

        JButton button = new JButton(BUTTON_CLICK_TITLE);
        button.setActionCommand(COMMAND_CLICK);
        add(button);

    @Override
    public void actionPerformed(ActionEvent e) {

        if (e.getActionCommand().equals(COMMAND_CLICK)) {
```