



# Medientechnik

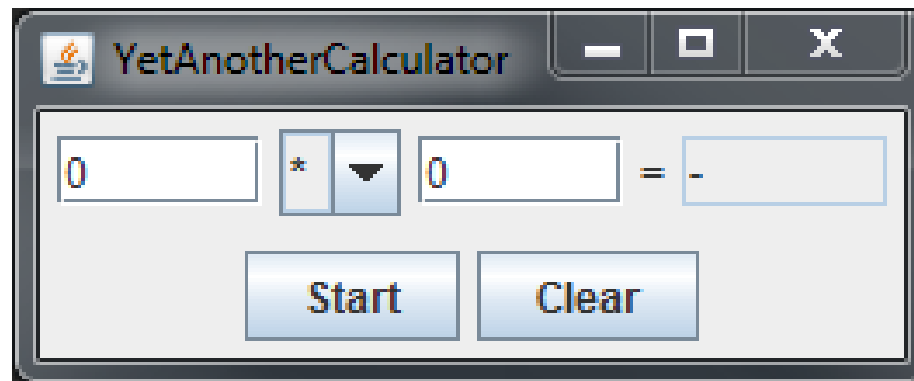
## Übung 2 – Java Swing

# Planung

Nr	Zeitraum	Thema
1	20.04. – 25.04.	Bildretusche mit Gimp
2	27.04. – 01.05.	GUI Programmierung
3	04.05. – 08.05.	Model-View Controller
4	18.05. – 22.05.	Bildfilter
5	26.05. – 29.05.	Video & Film Theorie
6	29.05. – 03.07.	Audio-Aufnahme und -Bearbeitung

# Heute

- GUI-Programmierung mit Eclipse
- Java Swing
  - Fenster erstellen
  - GUI-Komponenten hinzufügen
  - Layout-Manager nutzen



# Links zu Tutorials

Oracle:

<http://download.oracle.com/javase/tutorial/uiswing/>

Galileo Computing „Java ist auch eine Insel“:

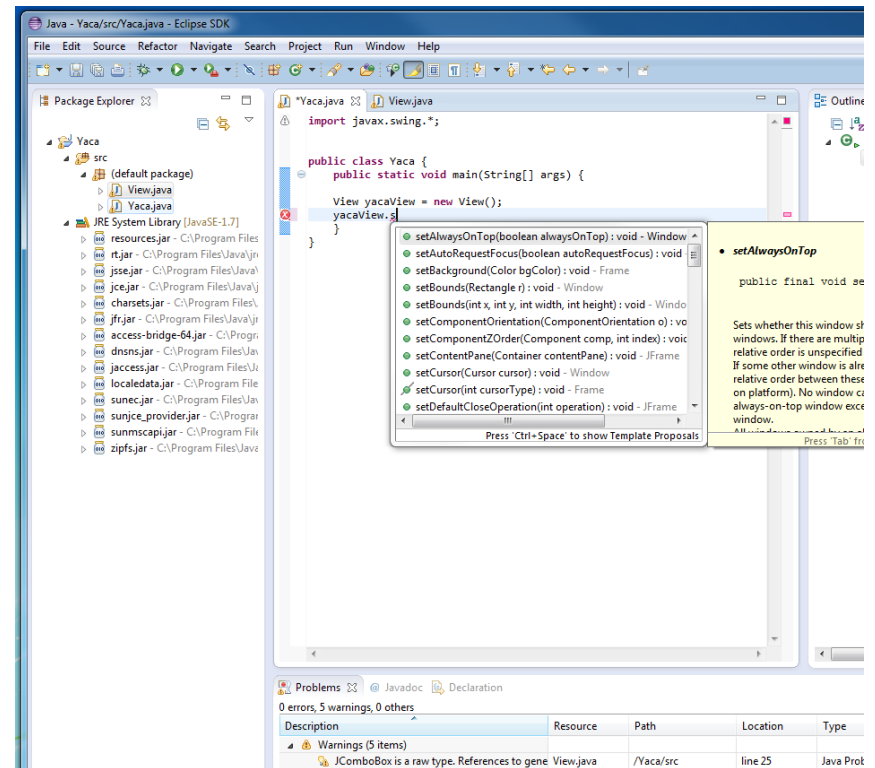
<http://openbook.galileocomputing.de/javainsel/>

Zetcode:

<http://zetcode.com/tutorials/javaswingtutorial/>

# Eclipse

Programmierwerkzeug  
(ursprünglich integrierte  
Entwicklungsumgebung für  
Java, mittlerweile auch für  
viele andere  
Programmiersprachen)



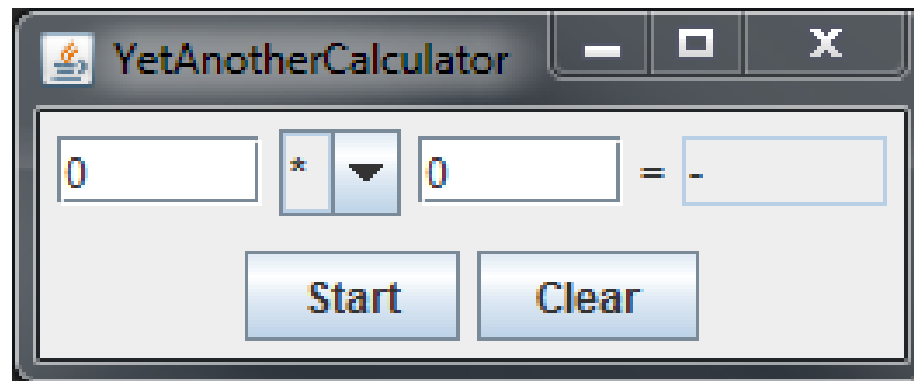
# Eclipse

- Vorteile:
  - Verschiedene Views
  - Syntaxhighlighting
  - Code Completion
  - Code-Folding
  - Und viele weitere
- Shortcuts (Beispiele):
  - Strg+Leertaste: Code Completion
  - Strg+Linksklick auf Funktionsaufruf:  
Sprung zur Deklaration der Funktion
  - Cursor über Variable: alle Vorkommen  
der Variable werden farbig unterlegt



# Taschenrechner

- Projekt in dieser und der nächsten Übung
- Heute:
  - Elemente
  - Layout



# Eclipse

- Eclipse starten und Workspace festlegen
- Neues Java-Projekt anlegen
- 2 neue Klassen im default-package erstellen:
  - `Yaca.java`
  - `View.java`



# Java – main-Methode

Yaca.java

```
public class Yaca {  
    public static void main(String[] args) {  
  
        View yacaView = new View();  
        yacaView.setVisible(true);  
    }  
}
```

# JFrame

View.java

```
import javax.swing.*;  
  
public class View extends JFrame {  
  
    public View() {  
        this.setTitle("YetAnotherCalculator");  
        this.setDefaultCloseOperation(  
            WindowConstants.EXIT_ON_CLOSE);  
    }  
}
```



# JPanel

View.java

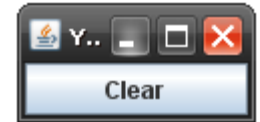
```
import javax.swing.*;  
  
public class View extends JFrame {  
  
    public View() {  
        this.setTitle("YetAnotherCalculator");  
        this.setDefaultCloseOperation(  
            WindowConstants.EXIT_ON_CLOSE);  
  
        JPanel contentAll = new JPanel();  
        contentAll.setLayout(new BorderLayout());  
    }  
}
```



# JButton

View.java

```
import java.awt.*;  
import javax.swing.*;  
  
public class View extends JFrame {  
  
    public View() {  
        [...]  
        JButton start = new JButton("Start");  
        JButton clear = new JButton("Clear");  
        contentAll.add(start, BorderLayout.SOUTH);  
        contentAll.add(clear, BorderLayout.SOUTH);  
  
        this.setContentPane(contentAll);  
        this.pack();  
    }  
}
```



# JButton

View.java

```
import java.awt.*;
import javax.swing.*;

public class View extends JFrame {

    public View() {
        [...]
        JButton start = new JButton("Start");
        JButton clear = new JButton("Clear");

        JPanel contentButtons = new JPanel();
        contentButtons.setLayout(new FlowLayout());
        contentButtons.add(start);
        contentButtons.add(clear);
        contentAll.add(contentButtons, BorderLayout.SOUTH);

        this.setContentPane(contentAll);
        this.pack();
    }
}
```



# JTextField & JComboBox

View.java

```
import java.awt.*;
import javax.swing.*;

public class View extends JFrame {
    public View() {
        [...]

        JTextField firstInput      = new JTextField(5);
        JTextField secondInput     = new JTextField(5);
        JTextField result          = new JTextField(5);
        String[] methods           = {"+", "-", "*", "/"};
        JComboBox methodBox       = new JComboBox(methods);
        this.setContentPane(contentAll);
        this.pack();
    }
}
```



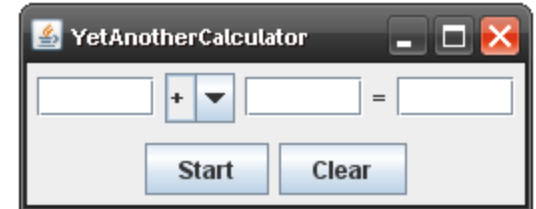
# JTextField & JComboBox View.java

```
import java.awt.*;
import javax.swing.*;

public class View extends JFrame {

    public View() {
        [...]
        JPanel contentInput = new JPanel(new BorderLayout());
        contentInput.add(firstInput);
        contentInput.add(methodBox);
        contentInput.add(secondInput);
        contentInput.add(new JLabel("="));
        contentInput.add(result);
        contentAll.add(contentInput, BorderLayout.CENTER);

        this.setContentPane(contentAll);
        this.pack();
    }
}
```



# default-Werte ändern

View.java

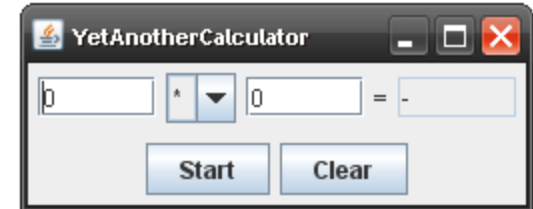
```
import java.awt.*;
import javax.swing.*;

public class View extends JFrame {

    public View() {
        [...]

        firstInput.setText("0");
        secondInput.setText("0");
        result.setText("-");
        result.setEditable(false);
        methodBox.setSelectedIndex(2);

        this.setContentPane(contentAll);
        this.pack();
    }
}
```





# Fenster-Position

View.java

```
import java.awt.*;
import javax.swing.*;

public class View extends JFrame {

    public View() {
        [...]

        this.setContentPane(contentAll);
        this.pack();

        this.setLocation(100, 100);
    }
}
```

# Wrap-up Quiz

1. Was ist ein Layout Manager?
2. Wofür wird ein Panel benötigt?
3. Was ist BorderLayout.SOUTH?
4. Was bewirkt  
.setDefaultCloseOperation()?
5. Welchen Punkt des Fensters bestimmt  
man mit setLocation()?
6. Was bewirkt pack()?



Vielen Dank!

**WELCHE FRAGEN GIBT ES?** 😊