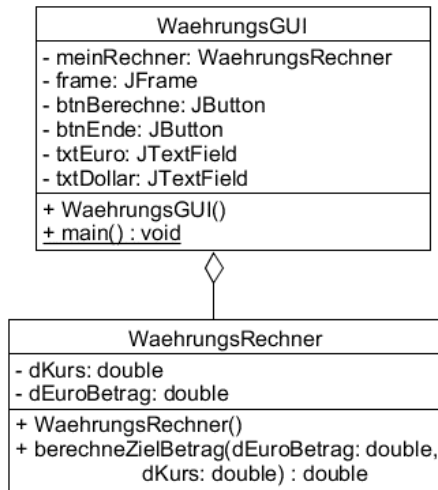


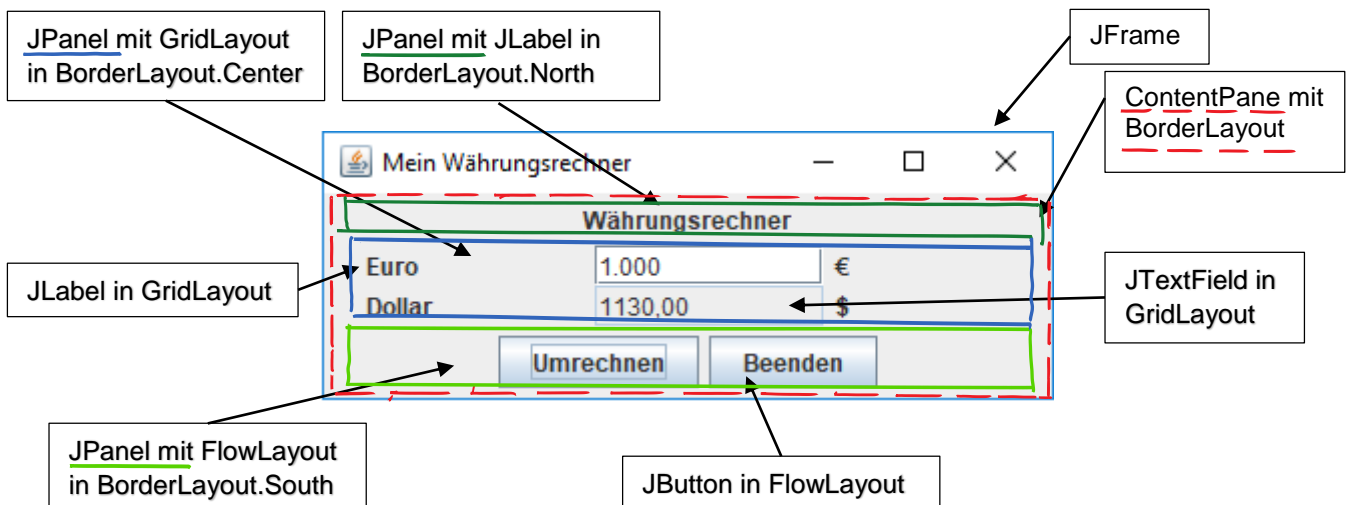
Einfache GUI Anwendung

Ziel: Erstellung einer einfachen GUI Anwendung: Kennenlernen des prinzipiellen Aufbaus eines JFrame mit JPanel, LayoutManagern und verschiedenen Dialogelementen.

1.1. Die zu erstellende Anwendung hat folgendes Klassendiagramm:



1.2. Die GUI der Anwendung hat folgende Struktur:



- 1.3. Erstellen Sie zunächst die Klasse WaehrungsGUI und zum Testen der GUI die Klasse WaehrungsStart.
- 1.4. Erstellen Sie eine MouseListener Klasse (abgeleitet von der MouseAdapter Klasse) für den „Beenden“ Button als innere Klasse von WaehrungGUI. In dieser Klasse muss nur die mouseClicked() Methode implementiert werden. Wenn der Button geklickt wird, soll das Programm mit dem Befehl `System.exit(0)` ; beendet werden.
- 1.5. Erstellen Sie eine MouseListener Klasse (abgeleitet von der MouseAdapter Klasse) für den „Umrechnen“ Button als innere Klasse von WaehrungGUI. In dieser Klasse muss nur die mouseClicked() Methode implementiert werden. Wenn der Button geklickt wird, geben Sie zum Testen zunächst nur einen Text im unteren JTextField aus.

1.6. Implementieren Sie im selben Package jetzt noch die Fachkonzept-Klasse

WaehrungsRechner.

1.7. Jetzt soll die `mouseClicked()` Methode von 1.5 abgeändert werden.

- aus dem Euro-Textfeld wird der String mit `getText()` ausgelesen.
- der String muss in einen double Wert konvertiert werden.
- dann kann mit einem Objekt der Fachkonzeptklasse der Zielbetrag berechnet werden.
- das Ergebnis wird mit `String.format()` für die Ausgabe im Dollar-Textfeld formatiert.

1.8. Wenn man das Programm testet, stellt man fest, dass es bei einer Fehleingabe im Euro-Textfeld möglicherweise mit einer Exception abstürzt. Aus diesem Grund ist es sinnvoller für dieses Textfeld statt der Klasse `JTextField` die Klasse

`JFormattedTextField` zu verwenden:

- ändern Sie die Verweisvariable von `JTextField` in `JFormattedTextField` ab
- erzeugen Sie das `JFormattedTextField` mit:
`new JFormattedTextField(NumberFormat.getNumberInstance())`
 → In diesem Textfeld können jetzt nur noch Zahlen eingegeben werden
- Der Wert aus dem Textfeld kann mit folgendem Befehl geholt und konvertiert werden:
`((Number)txtEuro.getValue()).doubleValue()`
- Testen Sie das geänderte Programm.

1.9. Eine weitere Verbesserung des Programms bestünde darin, dass die Umrechnung nicht nur dann ausgeführt wird, wenn man den Button klickt, sondern auch dann, wenn man im Euro-Textfeld die <Enter> Taste drückt. Hierfür gibt es selbstverständlich verschiedene Lösungen, eine sieht folgendermaßen aus:

Statt den beiden `MouseListener` Klassen wird eine `ActionListener` Klasse für die Ereignisbehandlung verwendet:

- Kommentieren Sie alles aus, was mit der `MouseListener` Ereignisbehandlung zu tun hatte.
- Definieren Sie als innere Klasse eine `ActionListener` Klasse die das `ActionListener` Interface implementiert:

```
private class BtnActionListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource().equals(btnEnde))
        {
            System.exit(0);
        }
        else
        {
            // hier die Berechnung durchführen
        }
    }
}
```

- Registrieren Sie den `ActionListener` jetzt sowohl auf die beiden Buttons als auch auf das Euro-Textfeld und testen Sie, ob das geänderte Programm, die Anforderung erfüllt.

```
public class WaehrungsRechner
{
    private double dKurs;
    private double dEuroBetrag;

    public WaehrungsRechner()
    {
        this.dEuroBetrag = 1;
        this.dKurs = 1.1;
    }

    public double berechneZielBetrag(double dEuroBetrag, double dKurs)
    {
        this.dEuroBetrag = dEuroBetrag;
        this.dKurs = dKurs;
        return dEuroBetrag * dKurs;
    }

    public double getdKurs()
    {
        return this.dKurs;
    }

    public double getdEuroBetrag()
    {
        return this.dEuroBetrag;
    }
}

* WaehrungsGUI.java
package guiApps;
import java.awt.*;
* @author stk
public class WaehrungsGUI
{
    private WaehrungsRechner meinRechner; // Verweis auf Fachkonzept Objekt
    // GUI Komponenten nur dann zu Attributen machen, wenn nötig
    private JFrame frame; // Verweis auf Fenster Objekt
    private JButton btnBerechne;
    private JButton btnEnde;
    private JFormattedTextField txtEuro;
    private JTextField txtDollar;

    public static void main(String[] args)
    {
        // ein Objekt der neuen GUI-Klasse erzeugen
        WaehrungsGUI fenster = new WaehrungsGUI();
        // Darstellung des Fensters in der optimalen Grösse
        fenster.frame.pack();
        // Darstellung des Fensters auf dem Bildschirm
        fenster.frame.setVisible(true);
    }
}
```

```

public WaehrungsGUI()
{
    meinRechner = new WaehrungsRechner();
    this.frame = new JFrame("Mein Währungsrechner");
    // Wenn das Fenster geschlossen wird, soll auch das Programm enden
    this.frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // In contentPane werden die Steuerelemente angelegt
    Container contentPane = frame.getContentPane();
    contentPane.setLayout(new BorderLayout());
    // Panel für den Titel
    JPanel pnlTitel = new JPanel(); // Standard ist FlowLayout
    JLabel lblTitel = new JLabel("Währungsrechner");
    pnlTitel.add(lblTitel);
    // Panel für die Textfelder und Labels
    JPanel pnlEinAusgabe = new JPanel();
    pnlEinAusgabe.setLayout(new GridLayout(0, 3));
    JLabel lblEuro = new JLabel("Euro");
    JLabel lblDollar = new JLabel("Dollar");
    JLabel lblEuroEinheit = new JLabel(" €");
    JLabel lblDollarEinheit = new JLabel(" $");
    this.txtEuro = new JFormattedTextField(NumberFormat.getNumberInstance());
    this.txtEuro.setValue(Double.valueOf(1));
    this.txtDollar = new JTextField(10);
    this.txtDollar.setEditable(false);
    pnlEinAusgabe.add(lblEuro);
    pnlEinAusgabe.add(this.txtEuro);
    pnlEinAusgabe.add(lblEuroEinheit);
    pnlEinAusgabe.add(lblDollar);
    pnlEinAusgabe.add(this.txtDollar);
    pnlEinAusgabe.add(lblDollarEinheit);
    // Panel für die Buttons
    JPanel pnlButtons = new JPanel(new FlowLayout());
    this.btnBerechne = new JButton("Umrechnen");
    this.btnEnde = new JButton("Beenden");
    pnlButtons.add(this.btnBerechne);
    pnlButtons.add(this.btnEnde);
    // Plazieren der Panels auf der Inhaltsebene (contentPane)
    contentPane.add(pnlTitel, BorderLayout.NORTH);
    contentPane.add(new JLabel(" "), BorderLayout.WEST);
    contentPane.add(pnlEinAusgabe, BorderLayout.CENTER);
    contentPane.add(pnlButtons, BorderLayout.SOUTH);

    // Auskommentiert Lösung für A1.4 / A1.5
    // Listener Objekt erzeugen und auf die Buttons registrieren
    // BtnBerechneListener btnBerechneListener = new BtnBerechneListener();
    // this.btnBerechne.addMouseListener(btnBerechneListener);
    // //btnBerechneListener.berechne();
    // this.btnEnde.addMouseListener(new BtnEndeListener());
    // Geänderte Lösung A1.9
    BtnActionListener btnActionListener = new BtnActionListener();
    this.btnEnde.addActionListener(btnActionListener);
    this.btnBerechne.addActionListener(btnActionListener);
    this.txtEuro.addActionListener(btnActionListener);
    berechne();
}

```



```
private void berechne()
{
    double dEuro = ((Number)txtEuro.getValue()).doubleValue();
    //double dEuro = Double.parseDouble(txtEuro.getText());

    double dDollar;

    dDollar = meinRechner.berechneZielBetrag(dEuro, 0.88);

    txtDollar.setText(String.format("%.2f", dDollar));
}

private class BtnBerechneListener extends MouseAdapter
{
    public void mouseClicked(MouseEvent e)
    {
        berechne();
    }
}

private class BtnEndeListener extends MouseAdapter
{
    public void mouseClicked(MouseEvent e)
    {
        System.exit(0);
    }
}

private class BtnActionListener implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource().equals(btnEnde))
        {
            System.exit(0);
        }
        else
        {
            berechne();
        }
    }
}
}
```