

**Projekt DM in Euro-Wechsler**

Im Laufe dieses Projektes soll ein Applet entstehen, das einen gegebenen DM Betrag in Euro wechselt und entsprechend der Euro-Stückelung passend ausgibt:

Liste: 2\*2Cent + 1\*5 Cent +...

Zentrales Lern-Thema dieses Projekts sind Verzweigungen und Programmdarstellung bzw. Dokumentation in Struktogrammen nach DIN 66261(Nassi Shneiderman TB Seite 192).

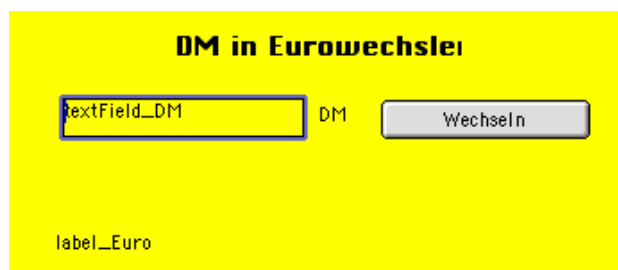
**Aufgabe 1 (Einfacher Umrechner, das EVA-Prinzip)**

Gegeben ist ein DM-Betrag, gesucht ist der entsprechende Euro-Betrag auf zwei Nachkommastellen gerundet (1€ = 1,95583 DM).

**E**ingabe: DM-Betrag  
**V**erarbeitung: Euro = DM / 1,95583  
 Euro auf zwei Nachkommastellen runden  
**A**usgabe: Euro-Betrag

**Schüler-Job:** Erstellen Sie ein Java-Projekt mit dem Namen „DM\_in\_Eurowechsler“ und ein Applet mit dem Namen „DM\_Euro“ das folgendes Aussehen hat:

**Achtung:** Kein Swing-Applet, die AWT verwenden, das Layout auf Null einstellen!



Benennen Sie den **Namen** des „textField1“ zu „textField\_DM“ um, **nicht** den Text.

**Info:** Label, Textfelder, Buttons usw. nennt man *Komponenten* des Applets. Wenn Sie ein Label in das Applet einfügen, wird eine *Objekt-Instanz* der *Objekt-Klasse* „Label“ erzeugt. Die Objekt-Klasse kann man sich als Bauplan für eine Objekt-Instanz vorstellen. Im Bauplan sind die *Attribute* (Eigenschaften) und *Methoden* (Verfahren) eines Objekts beschrieben. Attribute eines Labels sind beispielsweise sein Name, der Text auf ihm, die Position in der Umgebung usw.

Bei der *Instanzierung* eines Objekts werden die Attribute automatisch vorbelegt, *initialisiert*.

**Beispiel:** Ein neues Label bekommt den Namen und den Text „label2“.

Die Methoden eines Labels lauten z.B. getText(), setText(„Hallo Welt“) usw.

**Beispiel:** In das Euro-Label „Hallo Welt“ schreiben: label\_Euro.setText(„Hallo Welt“);

Mit dem Namen wird eine *Komponente* im Quelltext des Programms angesprochen. Namen von Komponenten, die im Quelltext verwendet werden sollten daher aussagekräftig sein!

Benennen Sie „button1“ in „button\_wechseln“ und „label3“ in „label\_Euro“ um.

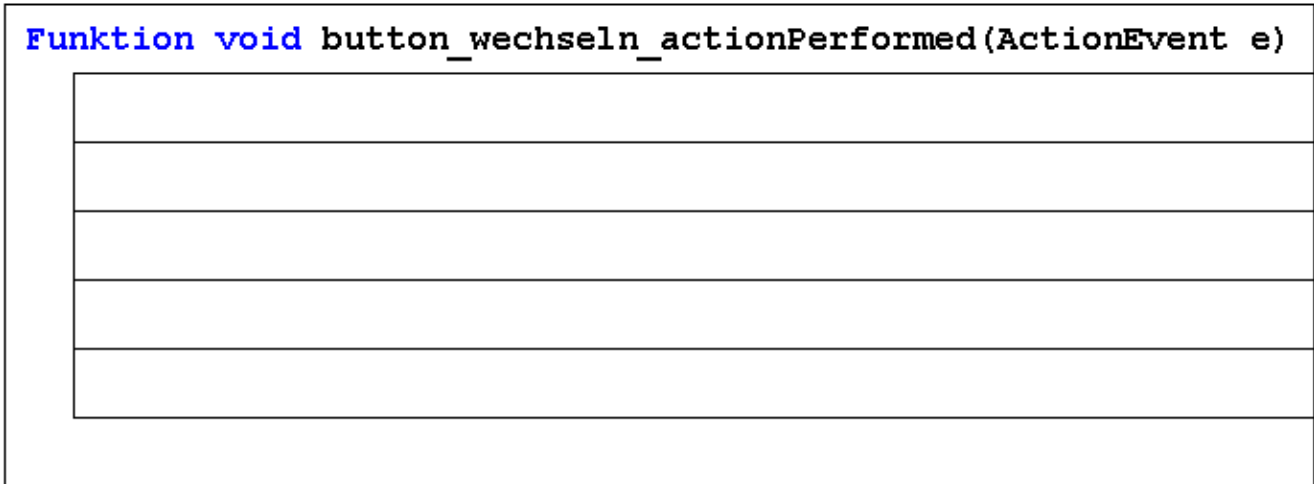
Bei einem Klick auf den „button\_wechseln“ soll die Umrechnung stattfinden, d.h. die Verarbeitung. Durch einen Doppelklick auf den Button wird im Quelltext folgender Funktionsrumpf erzeugt:

```
void button_wechseln_actionPerformed(ActionEvent e) {
}
}
```

In der Struktogrammdarstellung von EasyCODE(JAVA) V6.8 (befindet sich auf Ihrer TGIT-CD) sieht der Funktionsrumpf so aus:

**Gemeinsam:** Ich habe Platz für die fünf Prosa-Anweisungen gelassen, die wir nun einfügen...

### `button_wechseln_actionPerformed`



**Hinweis:** Mit EasyCODE kann man Programme auch mittels Struktogramm sehr komfortabel erstellen, leider ist die Demo-Version derart beschränkt, dass sich unsere Java-Projekte schon nicht mehr abspeichern lassen (Schachtelungstiefe ist beschränkt). Nebenbei haben Sie im Abi kein EasyCode zur Verfügung und die Struktogrammdarstellung entspricht nicht immer der Norm.

**Schüler-Job:** Erstellen Sie den Quelltext zu dem Struktogramm!

Verwenden Sie als Datentyp für DM und Euro „double“ (siehe Java-Script Seite 10)

Folgender Ausdruck liest aus Textfeld gibt einen double-Wert zurück:  
`Double.valueOf(textField_DM.getText()).doubleValue();`

Diese Funktion rundet nach dem Komma:  
`Math.round(99.5)` gibt 100.0 zurück

Diese Funktion wandelt double in String um:  
`Double.toString(100.34)` gibt "100.34" zurück

So können zwei Strings verbunden werden:  
"100.34" + " Euro" ergibt "100.34 Euro"

Der Text eines Labels wird so neu gesetzt:  
`label_Euro.setText("Neuer Text");`

**Ihre Lösung:**

```
void button_wechseln_actionPerformed(ActionEvent e) {  
  
  
  
  
  
  
  
  
  
}
```