



Graphik in Java



Bibliotheken

- AWT (Abstract Windowing Toolkit)
 - Package: `java.awt.*`
- Swing
 - Package: `javax.swing.*`

AWT

- ☉ plattformabhängig
- ☉ benutzt die GUI (Buttons, Fenster, ...) des BS
 - ✓ Look and Feel des Betriebssystems
 - ✓ schnell
 - minimaler Umfang

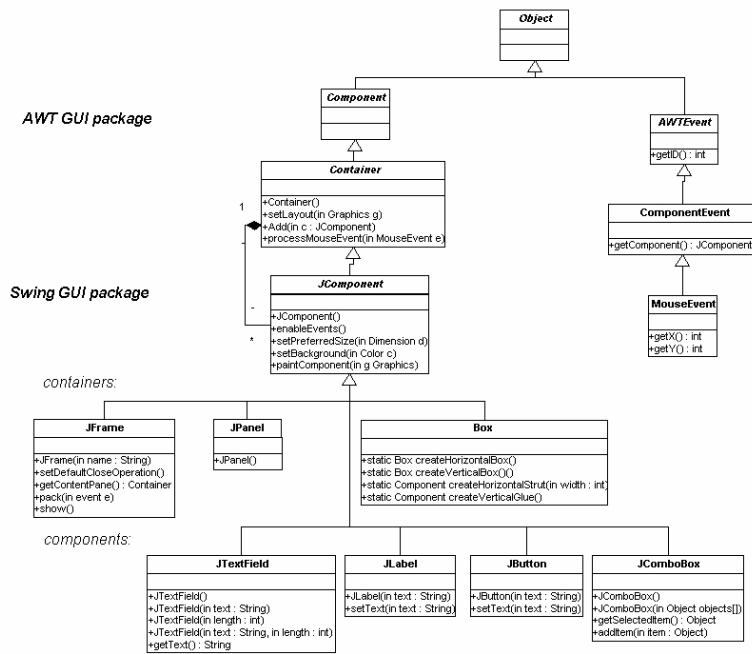
3

Swing

- ☉ plattformunabhängig
- ☉ GUI-Elemente werden selbst gezeichnet
 - ✓ Anwendungen sehen überall gleich aus
 - ✓ flexibel
 - ✓ umfangreiche Möglichkeiten
 - langsam

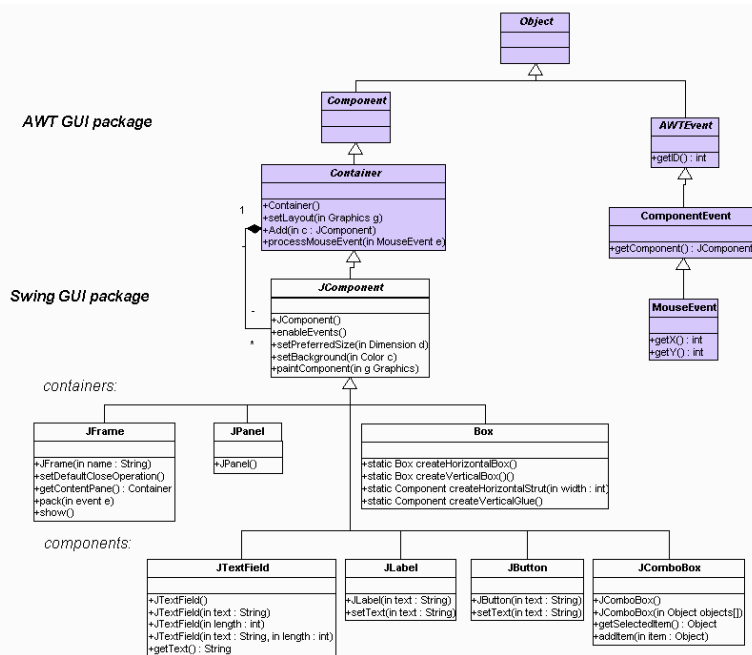
4

Klassenhierarchie



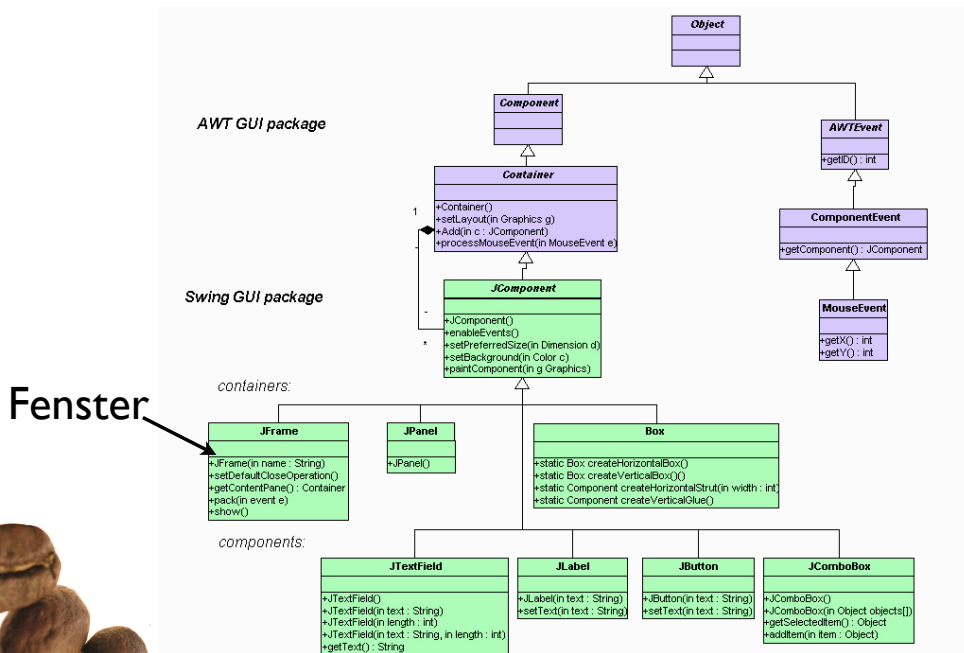
5

Klassenhierarchie



6

Klassenhierarchie



7

Das Hauptfenster

☪ Klasse: `javax.swing.JFrame`

```
public class GUI extends JFrame {

    public GUI() {
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setTitle("Dijkstra-Algorithmus");
        this.setSize(634, 769);
    }

    public static void main(String[] argv) {
        GUI ui = new GUI();
        ui.setVisible(true);
    }
}
```

8

Graphik-Kontext

- Zeichnen von Graphik-Primitiven
- Auf alles, was im Fenster sichtbar ist
- Klasse: `java.awt.Graphics`
- Attribut von `java.awt.Container`

9

Zeichnen auf einen Container

- Fläche, auf die gezeichnet werden soll:
 - `Container cont`
- Graphik-Kontext dieser Fläche:
 - `Graphics g = cont.getGraphics();`
- Zeichnen auf die Fläche:
 - `g.zeichnenMethode();`

10

Zeichenmethoden

- Eigenschaften der Linien:

- `g.setColor(Color.black);`

- Zeichnen von Linien:

- `g.drawLine(...), g.drawString(...)`

- `g.drawOval(...), g.drawRect()`

- Zeichnen von Flächen:

- `g.fillOval(...), g.fillRect()`

11

Zeichenfläche des Hauptfensters

- Innenfläche des Fensters: `ContentPane`
(vom Typ `java.awt.Container`)

- erreichbar über `get`-Methode:
`getContentPane()`

- Graphik-Kontext der Innenfläche:
`getContentPane().getGraphics();`



12

Koordinatensystem



13

Fragen?