

2D Spiele programmieren in Java

Teil 1: Grundlagen und
Spielidee

Dr. Katja Wegner
Dr. Ursula Rost

2D Spiele programmieren

- Bevor es losgeht: ÜBERLEGEN
 - Was soll das Spiel haben und verwenden?
 - Was soll das Spiel tun und wie oft?
 - Wie sollte die passende Klassenstruktur aussehen?



Was Spiel haben können

- **GUI** – graphische Oberfläche (Titel, Menüs)
- **Graphik** (Hintergrund, Animationen)
- **Sound** (Hintergrundmusik)
- **Objekte** (Figuren, Punktestände, Zeit, Leben)
- **Objekt-Container** (Levels, Karten, Spielflächen)
- **Events** (Eingaben des Spielers)

Was Spiele tun können

- Pro Spiel/Level:
 - Laden graphische Objekte und Sound
 - Erzeugen Objekte und Container
 - Besetzen der Container mit Objekten
- **Ständig (Spielroutine):**
 - Eingaben verarbeiten
 - Objekte aktualisieren
 - Grafiken anzeigen
 - Sound abspielen

Spielidee Muka

- Mukas bewegen sich über ein Spielbrett und versuchen ihren Feinden auszuweichen und Nahrung einzusammeln.



Umsetzung



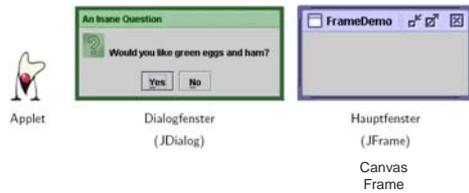
- Spielbar als **Applet** oder in der
- **Konsole:**
 - Vollbild oder
 - Fenster mit bestimmter (fester) Größe

Grafische Benutzeroberfläche

- **AWT** (Abstract Window Toolkit) und Swing
 - Grafikbibliothek
 - Komponenten zum Bilden von grafischen Benutzerschnittstellen
 - Fenster, Knöpfe, Menüs
 - Event listeners zum Behandeln von Benutzereingaben
 - Java 2D:
 - Hinzufügen von 2D Grafiken, Bildern oder Text
 - Drag & Drop Unterstützung

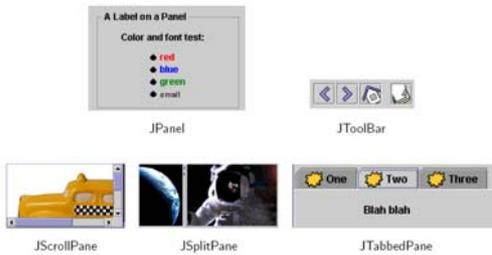
Top-Level Container

- Stehen an oberster Stelle der Hierarchie

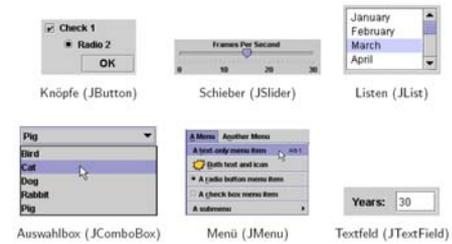


- Alles mit J beginnend gehört zu Swing ohne zu AWT !!!

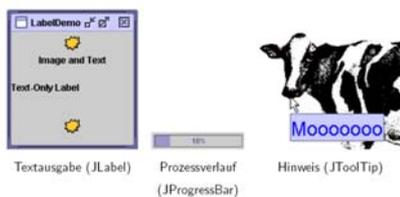
Zwischenlevel Container



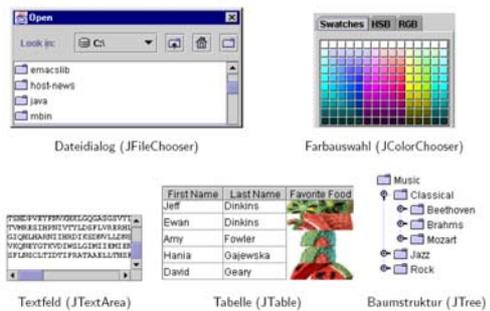
Kontrollelemente



Informationen anzeigen



Interaktive Komponenten



Basisstruktur eines Programms

```
import javax.swing.*;
public class MyGui extends JFrame {
    public MyGui() {
        super("Name meiner GUI"); // call constructor of JFrame
    }

    public static void main(String[] args) {
        MyGui gui = new MyGui(); // Objekt von MyGui
        gui.initFrame(); // Aufruf der Initialisierungsmethode
    }

    public void initFrame() {
        ... // Komponenten hinzufügen
        this.setVisible(true); // Anzeigen der Oberfläche
    }
}
```

Hinzufügen von Komponenten

```
import javax.swing.*;
public class MyGui extends JFrame {
    public MyGui() {
        super("Name meiner GUI"); // call constructor of JFrame
    }

    public static void main(String[] args) {
        MyGui gui = new MyGui(); // Objekt von MyGui
        gui.initFrame(); // Aufruf der Initialisierungsmethode
    }

    public void initFrame() {
        ... // Komponenten hinzufügen
        this.setVisible(true); // Anzeigen der Oberfläche
    }
}
```

Übung 1

Aufgabe:

- Probieren Sie verschiedene Methoden von Graphics aus und zeichnen Sie 10 verschiedene Objekte in verschiedenen Farben an verschiedenen Positionen.
- Beispielprogramm: DrawFrame.java