Ludwig Maximilians Universität München

Java - SWING

Die

SWING

Klassenbibliothek

Ludwig Maximilians Universität München

Inhalt

- AWT vs. SWING
- Struktur und Praxis
  - MVC-Pattern
  - Erweiterte und komplexere Komponenten
  - GUI-Architektur
  - Threadsicherheit

Ludwig Maximilians Universität München

LMU

Allgemeines

- "Lightweight"-Components
- VM erzeugt Graphik
- Nur ein bemalbares Rechteck wird benötigt
- Oberfläche sehr gut portierbar

Ludwig Maximilians Universität München

MVC-Pattern

Trennung von Datenmodell und Darstellung

- Jede Swing-Komponente besteht aus
  - Modell (Daten)
  - View (UI-graphische Oberfläche)
  - Controller
- dh.: UI ist austauschbar...

Ludwig Maximilians Universität München

Klassenhierarchie

- Eingebettet in AWT (Paket: `javax.swing`)
- Nutzung und Erweiterung des Event-Mechanismus
- Graphische Elemente (JButtons, JLabels, JTextfelder etc.)
- Komplexe Elemente (JColorChooser, JFileChooser)

Ludwig Maximilians Universität München

LMU

Neuer Leistungsumfang

- Austauschbares Look and Feel
- Drag and Drop
- Internationalisierung

Ludwig Maximilians Universität München

Schichtenstruktur eines JFrame

- Frame
  - RootPane
  - LayeredPane
  - ContentPane
  - GlassPane

  Zugriff jeweils durch `get`- und `set`-Methoden

  Siehe Beispiel `GlasPane`

Ludwig Maximilians Universität München

Threadsicherheit

- Swing ist **nicht** synchronisiert!
  - Graphikarchitektur in SW-Design einbeziehen!
  - Graph. Operationen in die Queue des Swing-Threads stellen

  zB.: durch `invokeLater(Runnable r)`

  Siehe Beispiel `GlasPane`
- Jede Swinganwendung ist an sich nebenläufig!

Ludwig Maximilians Universität München

Threadsicherheit

- Swing verfügt über einen "EventDispatchingThread"
- Erzeugung, Modifikation und Abfrage graphischer Elemente nur von hier.
  - `invokeLater`: der Kontrollfluß des aufrufenden Threads läuft sofort weiter.
  - `invokeAndWait`: der Kontrollfluß hält. (zB.: modale Dialoge)

Ludwig Maximilians Universität München

Actions

- Eventhandlercode wird mehrfach benötigt
- Integration in eine `Action`
- Action reduzieren Redundanz
  - Bezeichnungen (Textlabels)
  - Icons
  - Tooltips
  - Funktionalität
  - Hotkeys/Shortcuts

Ludwig Maximilians Universität München

Zusammenfassung

1. Klassische Entwicklung der Applikation (Attribute, Methoden)
2. Design der graphischen Oberfläche
3. Entwicklung der Eventlistener
   - Aufruf spezieller Methoden vermittelt durch Events
   - Evtl. Entwicklung von `Actions`
4. Update der, bei der Programmbearbeitung veränderten Werte
   - Modell-View-Controller Architektur