

Einführung in die Programmierung für NF

Übung 09

18.12.2013

Inhalt

- Korrektur Blatt 8
- Wiederholung Design Patterns
- Einführung in GUI's mit Java Swing und AWT

Korrektur Blatt 8

- Musterlösung wieder auf der Homepage zum Download (ohne Listen-Funktionen)
- Kann zur Vorlage für Blatt 9 dienen

Wiederholung Design Patterns

- MVC Pattern
- Logische Trennung von Model, View und Controller
- Model: Anwendungslogik
- View: Sichtbares User Interface
- Controller: Kommunikation zwischen Model und View, Start der Applikation

Wiederholung Design Patterns

- Observer Pattern
- Trennung von Aktion und Reaktion
- Ein Observer überwacht ein Observable
- In unserem Fall:
 - Model ist ein Observable
 - View ist ein Observer und beobachtet das Model
 - View ändert/aktualisiert ihre Anzeige, wenn sich die Daten im Model ändern

Java AWT

- Das Abstract Window Toolkit (AWT) ist eine API zur Entwicklung von grafischen Oberflächen in Java
- AWT ist plattformunabhängig und verfügt daher über einen sehr beschränkten Umfang
- Umfang erweiterbar durch Java Swing

Java Swing

- Swing ist die Standardbibliothek für grafische Oberflächen in Java und erweitert das AWT
- Sie verfügt über Zeichenfunktionen grafischer Grundelemente und Farben
- Sie bietet grafische Steuerelemente
- Sie definiert ein Modell zur Behandlung von Ereignissen wie z.B. Mausklicks

http://openbook.galileocomputing.de/javainse19/javainse19_001.htm

Erstellung eines Fensters

- Oberklasse JFrame

```
JFrame f = new JFrame("Titel");
```

```
Public class View extends JFrame { }
```

- Erzeugt einen Rahmen (Frame), in dem sich der restliche Inhalt befindet

http://openbook.galileocomputing.de/javainsel9/javainsel_19_003.htm

Erstellung eines Fensters

- JFrame beinhaltet mehrere wichtige Standardfunktionen:
 - setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 - setSize(int width, int height);
 - setVisible(boolean visible);
- Beispiel:

```
JFrame f = new JFrame("Beispiel");  
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
f.setSize(300, 200);  
f.setVisible(true);
```

Java Swing Komponenten

- In einem JFrame werden nun die weiteren Elemente platziert
 - JPanel als Container für mehrere Elemente
 - JLabel für Beschriftungen und Text allgemein
 - JList für Listendarstellungen
 - JButton für Schaltflächen
 - JSlider, JScrollbar, JCheckbox, JProgressBar, JComboBox, JSpinner u.v.m.

Platzierung von Komponenten

- Hinzufügen eines Elements:

```
JFrame frame = new JFrame("Beispiel");  
JLabel label = new JLabel("Irgendein Text");  
frame.add(label);
```

oder mit Container

```
JFrame frame = new JFrame("Beispiel");  
JPanel panel = new JPanel();  
JLabel label = new JLabel("Irgendein Text");  
panel.add(label);  
frame.add(panel);
```

Verwendung von Layoutmanagern

- Containern oder Elemente können so nur (nacheinander) platziert werden
- Für die Umsetzung von Layouts gibt es eine Vielzahl an Layoutmanagern
- Hier relevant sind FlowLayout und BorderLayout

http://openbook.galileocomputing.de/javainasel9/javainasel_19_011.htm

FlowLayout

- Das FlowLayout ordnet die platzierten Elemente einfach der Reihe nach an
- Es ist sehr einfach zu implementieren, denn es muss nur als Layout festgelegt werden

FlowLayout

```
JFrame frame = new JFrame("Beispiel");  
frame.setLayout( new FlowLayout() );  
frame.add( new JLabel ("Hallo") );  
frame.add( new JLabel ("Welt") );  
frame.pack();  
frame.setVisible(true);
```

BorderLayout

- Das BorderLayout verfügt über 5 Positionen: Die vier Himmelsrichtungen und die Mitte (Center)
- Die Elemente im Norden und Süden richten sich in der Höhe am Inhalt und erstrecken sich über die gesamte Fensterbreite
- Die Elemente in Westen und Osten werden von der Höhe passend gestreckt, die Breite kann definiert werden
- Das Element in der Mitte wird immer angepasst

BorderLayout

- Die Position wird in der add()-Funktion mit übergeben, bleibt der Parameter leer, wird automatisch in der Mitte platziert

```
frame.setLayout( new BorderLayout() );
```

```
frame.add( new JButton(), BorderLayout.NORTH );
```

```
frame.add(new JButton(), BorderLayout.WEST );
```

```
frame.add(new JButton(), BorderLayout.CENTER );
```

```
frame.add(new JButton(), BorderLayout.EAST );
```

Verwendung von Layoutmanagern

- JPanels können ebenfalls eigene Layoutmanager zur Anordnung des Inhalts bekommen
- Alle Elemente (besonders JPanels) können beliebig ineinander verschachtelt werden
- Dringende Video-Empfehlung:
<https://www.youtube.com/user/sep1213>

Fragen zum Übungsblatt?