Prof. Kastens und Mitarbeiter

Vorlesung Software-Entwicklung II SS 2004 - Lösung zu Blatt 2

Lösung zu Aufgabe 3

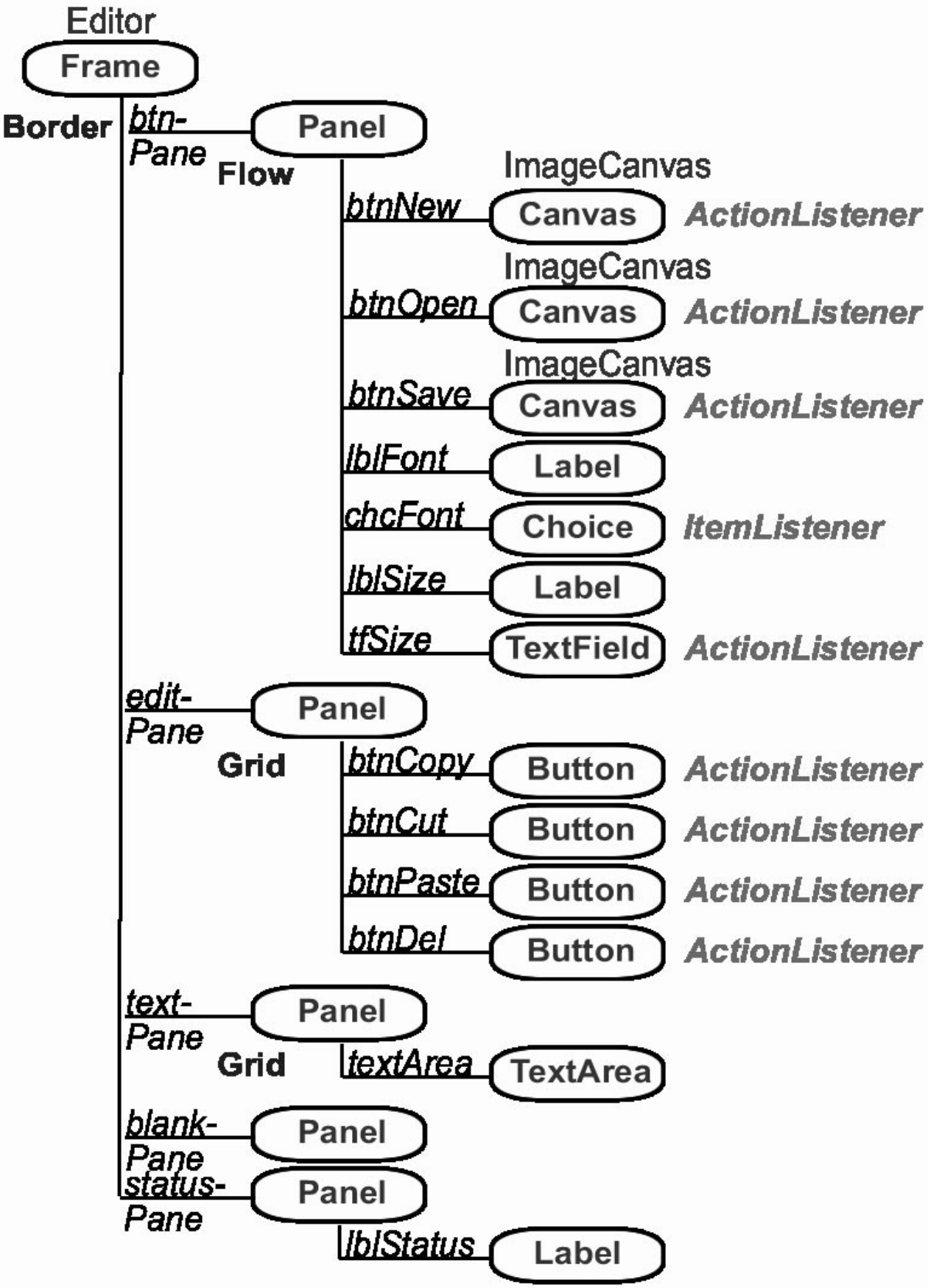
Eigenschaften, die gegenüber Ändern der Ausmaße des Containers invariant sind:

- **BorderLayout**
  - Bis zu 5 rechteckige Flächen jeweils mit einer Komponente werden angezeigt.
  - Anordnung: NORTH über allen, SOUTH unter allen, WEST links von CENTER, EAST rechts von CENTER.
  - NORTH und SOUTH nehmen die ganze Breite des Containers ein, die Höhe wird von den Komponenten bestimmt.
  - WEST und EAST nehmen die verbleibende Höhe ein, Breite wird von den Komponenten bestimmt.
  - CENTER bekommt die verbleibende Höhe und Breite.
- **FlowLayout**
  - Reihenfolge der Komponenten bleibt unverändert.
  - Komponenten werden in Zeilen angeordnet, sodass alle Zeilen außer der letzten maximal die Breite des Containers füllen.
  - Die Höhe jeder Zeile wird durch die maximale Höhe ihrer Komponenten bestimmt.
  - Alle Zeilen werden nach links oder rechts ausgerichtet oder zentriert.
- **GridLayout**
  - Die Flächen werden als Matrix mit fester Anzahl von Zeilen und Spalten angeordnet.
  - Alle Flächen sind Rechtecke gleicher Größe.
  - Die Flächen füllen den Container maximal aus.

Lösung zu Aufgabe 4

a) In dieser Aufgabe ging es zunächst darum, einen Objektbaum zu entwerfen.

So könnte der Baum aussehen:


Editor
Frame
Border
btn-Pane
Panel
Flow
ImageCanvas
btnNew
Canvas
ActionListener
ImageCanvas
btnOpen
Canvas
ActionListener
ImageCanvas
btnSave
Canvas
ActionListener
lblFont
Label
chcFont
Choice
ItemListener
lblSize
Label
tfSize
TextField
ActionListener
edit-Pane
Panel
Grid
btnCopy
Button
ActionListener
btnCut
Button
ActionListener
btnPaste
Button
ActionListener
btnDel
Button
ActionListener
text-Pane
Panel
Grid
textArea
TextArea
blank-Pane
Panel
status-Pane
Panel
lblStatus
Label

b) **1. Komponentenmethoden:**

- public Dimension getPreferredSize();
- public Dimension getMinimumSize();

2. Methoden der Schnittstelle LayoutManager:

- Dimension preferredLayoutSize(Container)
- Dimension minimumLayoutSize(Container)

3. Berücksichtigung der Größe:

- i. BorderLayout ignoriert die Größe der Komponenten teilweise: Höhe der nördlichen und südlichen Komponente wird berücksichtigt, Breite ignoriert. Breite der östlichen und westlichen Komponenten wird berücksichtigt, Höhe ignoriert. Höhe und Breite der zentralen Komponente werden ignoriert.
- ii. FlowLayout: Berücksichtigt die bevorzugte Breite und Höhe der Komponenten.
- iii. GridLayout: Ignoriert die bevorzugte Breite und Höhe der Komponenten.

Lösung zu Aufgabe 5

Für eine Beispiellösung siehe `Memory.java`.

Wir implementieren eine ActionListener-Klasse als innere Klasse der Hauptklasse, damit wir aus der Reaktionsmethode (actionPerformed) auf Zustandsvariable zugreifen können. Dies sind Variable, die die bisher aufgedeckten Felder speichern. Die Methode kann aufgerufen worden sein, wenn kein, ein, oder zwei Felder aufgedeckt sind:

- kein Feld: neues Feld wird angezeigt.
- ein Feld: neues Feld wird angezeigt; prüfen, ob die Marken gleich sind: wenn ja, beide kennzeichnen und für eine neue Wahl vorbereiten; wenn nein, auf nächste Aktion warten.
- zwei Felder: die vorher aufgedeckten (unterschiedlich markierten) Felder verdecken; das neue anzeigen.

Das Verdecken eines ungleichen Paares können wir später auch nach einer Verzögerungszeit auslösen, statt bis zur nächsten Auswahl zu warten.

Wir benötigen eine spezialisierte Button-Unterklasse, die ihre Beschriftung speichern, verdecken und wieder aufdecken kann.

Hier finden Sie die Folien aus der Zentralübung `zue2.pdf`