

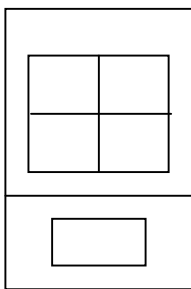
1 a)

```

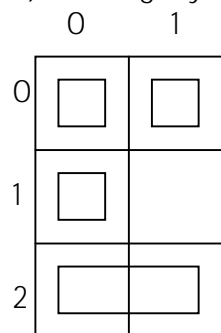
JButton JB1 = new JButton („Anfang“)
Jbutton JB2 = new JButton (“Ende“)
JButton JB3 = new JButton (“Reset“)
JTextField JT1 = new JTextField();
    
```

b)

1) Flowlayout



2) GridBagLayout



Codebeispiel:

```

JPanel P2 = new JPanel();
P2.setLayout(new GridBagLayout(2,2));
P2.add(JB1);
P2.add(JB2);
P2.add(JB3);
    
```

```

Contentpane.add(P2);
ContentPane.add(JT1);
ContentPane.setLayout(new GridBagLayout(2,1));
pack();
    
```

```
Contentpane.SetLayout(newGridBagLayout());
GridBagConstraints gc = new GridBagConstraints();
```

```
gc.gridx = 0;
gc.gridy = 0;          top.left,bottom,right
gc.insert = new inserts(4,4,4,4);
ContentPane.add(JB1,gc);
```

```
gc.gridx = 1;
gc.gridy = 0;
ContentPane.add(JB2,gc);
gc.gridx = 0;
gc.gridy = 1;
ContentPane.add(JB3,gc);
gc.gridx = 0;
gc.gridy = 2;
gc.gridwidth = 2;
gc.fill = gc.HORIZONTAL;
ContentPane.add(JT1,gc);
pack();
```

c)

```
public class GoTest extends JFrame implements ActionListener
```

```
·
·
·
```

```
JB1.SetActionCommand("<p>");
JB1.SetActionCommand("</p>");
JB1.addActionListener(this);
JB2.addActionListener(this);
reset();
JB3.addActionListener(new ActionListener());
{
public void action Performed(ActionEvent e)
{
reset();}
}
pack();
```

```
public void actionPerformed(ActionEvent e)
{
JT1.setText(JT1.getText() + e.getActionCommand());
JB1.setEnabled(!JB1.getEnabled());
JB2.setEnabled(!JB2.getEnabled());
}
```

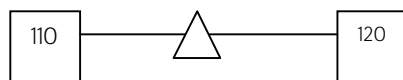
```

public void reset()
{
JT1.setText("");
JB1.setEnabled(true);
JB2.setEnabled(false);
}
}
2)

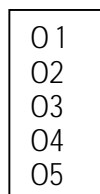
```

a) Liste, Radiobutton, Textfield

Intervall: Jslider nur Zahlenbereich liegt fest



1 aus n: JRadioButtonGroup



diskrete, festgelegte Anzahl

bel. Wert: JTextField „Freie Eingabe“

b)

JTextField tf;

intvalue = Integer.parseInt(tf.getText());

3

a)

Alle Listener = InterfaceKlassen(Schnittstellen) -> alle Meth. müssen impl. sein!
Adapter: abstrakte Klasse (extends) Nicht alle Methoden impl. -> Zwischenklasse

b) Kaskade: Abfrage von Bedingung. Erste Methode prüft die Ereignisse
Delegation: „Übertragen“ zu jedem Objekt gibt es eine entsprechende Methode

- c) - entsprechende Klasse wird von JDialog abgeleitet.
- Instanz selber muss modal gesetzt werden (setModal(true)) oder Super(owner,true);
- d) ActionEvent e
- e) public void actionPerformed(ActionEvent e)
- f) Label kann sich neuzeichnen. Drawstring müsste selbst immer wieder neu erstellt werden.
- g) 1 aus n -> abstrakt, da nichts vorgegeben

im abstrakten Layout werden prinzipiell die Interaktionen aufgezählt

Im konkreten Layout werden die Swing Objekte ausgeführt.
- h) - Fenstergröße wird verändert
- Beziehungen (z.B. Label oder Button werden verändert)
- Schriftgröße wird verändert
- i) BorderLayout: Center wird maximiert
GridLayout: alle Zellen gleich groß
FlowLayout: Objekte werden gemäß Einfügereihenfolge dargestellt (evtl. Umbruch).
- j) Bsp.:
JButton.setFont(auswahl);
ContentPane.setFont(new Font(...));
- k) - Tooltip
- Statusleiste
- Veränderung des Mausursors
- Fokussteuerung aktiv vermeiden
- Bedienelemente sichtbar / unsichtbar machen. Disable / enable

Fachhochschule Darmstadt Entwurf und Realisierung grafischer Oberflächen Fachbereich
Informatik Prof. Dr. Hans-Peter Wiedling

24.01.20
03
Klausur

Name	Vorname	Matrikel-Nummer	Note
------	---------	-----------------	------

Aufgabe 1	max. Punkte	erreichte Punkte
2	9	
3	22	
Praktikum	10	
gesamt	60	

Bemerkungen:

- Als Hilfsmittel ist das Skript zur Vorlesung erlaubt.
- Antworten Sie kurz und prägnant
- In Klammer ist die jeweils erreichbare Punktzahl angegeben.
- Arbeiten Sie gewissenhaft und sorgfältig

1. Aufgabe (29 Punkte)

Gegeben ist folgender Screen:



- a) (2) Erzeugen Sie alle Komponenten, die in dem Beispiel verwendet werden.
- b) (15) Machen Sie zwei prinzipiell unterschiedliche Vorschläge zum Layout; Skizzieren Sie diese und geben Sie den entsprechenden Code in Java an. Bemerkung: Gehen Sie davon aus, dass Ihnen die Variable `contentPane` vom Typ `JPanel` zur Verfügung steht.

Die Graphische Oberfläche verhält sich wie folgt:

Wird der Button mit der Beschriftung „Anfang“ gedrückt, dann wird an den Inhalt des Eingabefeldes die Zeichenfolge „<p>“ angefügt; beim Anklicken des Buttons „Ende“ wird „</p>“ angefügt. Der Reset-Button setzt beim Anklicken das Eingabefeld auf leeren String zurück.

- c) (10) Realisieren Sie vollständig das Eventhandling; Die Ereignisbehandlung der Buttons Anfang und Ende werden von einer Methode realisiert.
- d) (2) Ergänzen Sie Ihren Vorschlag um eine Benutzerführung: es soll nur möglich sein, den String „</p>“ anzufügen, wenn zuvor „<p>“ erzeugt wurde

2. Aufgabe (9 Punkte)

Geben Sie drei prinzipiell unterschiedliche Möglichkeiten an, eine Integerzahl mit Hilfe von Instanzen der Java Swing-Klassen einzugeben (der Wert steht dann in der `int`-Variable `value`):

- a) (6) Benennen Sie diese und erstellen Sie jeweils eine einfache Skizze. Welche Anwendungsaufgabe sehen Sie jeweils?
- b) (3) Geben Sie den benötigten Programmcode in Java/Swing zu einer der Möglichkeiten an

3. Aufgabe (22 Punkte)

Beantworten Sie die folgenden Fragen knapp aber präzise:

- a) (2) Was ist der Unterschied zwischen einer Listener- und einer Adapterklasse?
- b) (2) Erklären Sie im Kontext des Eventhandlings die Begriffe Kaskade und Delegation
- c) (2) Was ist notwendig, um einen modalen Dialog zu erstellen?
- d) (1) Welches Ereignis wird erzeugt, wenn der Benutzer einen Button anklickt?
- e) (1) Welche Methode wird aufgerufen, wenn der Benutzer einen Button anklickt?
- f) (2) Mit der Methode `drawString (...)` kann man einen Text an einer Position zeichnen. Geben Sie Vor- /Nachteile im Gegensatz zur Verwendung eines Labels an
- g) (1) Was ist der Unterschied zwischen konkretem und abstraktem Layout?
- h) (3) Nennen Sie 3 Situationen in denen bei einem dynamischen Layout das Layout neu berechnet wird
- i) (3)

Nennen Sie die Layoutstrategie beim

- $\frac{3}{4}$ BorderLayout
- $\frac{3}{4}$ GridLayout
- $\frac{3}{4}$ FlowLayout

j)

(2) Welche Aktionen sind notwendig, damit verschiedene Komponenten den gleichen Font verwenden können

k) (3) Nennen Sie drei unterschiedliche Möglichkeiten, dem Benutzer kontext- bzw. objektspezifische Zusatzinformation/Unterstützung zu geben