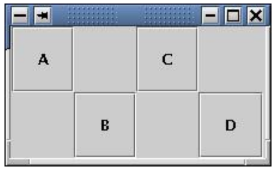
Dr. Johannes Mayer
Hans Braxmeier
Ralph Guderlei
Abteilung Angewandte Informationsverarbeitung
10. Juni 2005
Blatt 6

Design und Architektur von Softwaresystemen (SS 2005)

Abgabetermin: 17. Juni 2005

1 GridBagLayout (3 Punkte)

Befassen Sie sich mit dem *java.awt.GridBagLayout* und verwenden Sie es, um folgendes Layout zu erzeugen:



Bei Vergrößerung oder Verkleinerung des Fensters wachsen/schrumpfen die einzelnen Elemente (Buttons) entsprechend.

Erstellen sie ein Java-Swing-Programm unter Verwendung von GridBagLayout und Buttons, das dieses Layout erzeugt. Erklären Sie Ihrem Tutor die Klasse GridBagLayout und die Klasse GridBagConstraint.

2 ZigZagLayout (3 Punkte)

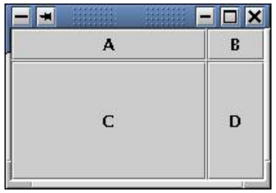
Implementieren Sie nun einen *eigenen Layout-Manager*, der die Buttons auf die selbe Weise wie in der ersten Aufgabe anordnet – im Zick-Zack abwechselnd in der ersten und zweiten Zeile.

Ihr Layout-Manger namens *ZigZagLayout* soll mit dem Programm *TestZigZag.java* verwendet werden können und das erwartete Resultat produzieren.

3 TableLayout (4 Punkte)

Implementieren Sie nun noch einen etwas sinnvolleren *eigenen Layout-Manager* names *TableLayout*, bei dem man die Aufteilung des freien Platzes für die Spalten und Zeilen über Stretch-Faktoren einstellen kann. Der Standard-Stretch-Faktor ist 0. Der verfügbare freie Platz wird proportional zu den Stretch-Faktoren der Zeilen und Spalten verteilt. Ansonsten werden die Komponenten analog zum GridLayout von links oben nach rechts unten angeordnet, wobei bei Konstruktor nur die Spaltenanzahl angegeben werden muss.

Das Beispielprogramm *TestTable.java* soll mit Ihrem Layout-Manager *TableLayout* lauffähig sein und folgendes Ergebnis produzieren:



Hinweise: Zur Implementierung der beiden Layout-Manager können Sie ja GridBagLayout verwenden. Man muss das Rad ja nicht zum x-ten Mal neu erfinden. ;-)

Die Methode *public void addLayoutComponent(String name, Component comp)* und die Methode *public void removeLayoutComponent(Component comp)* müssen Sie nicht implementieren. Sie können eine *UnsupportedOperationException* werfen. Bei *public void addLayoutComponent(Component comp, Object constraints)* müssen Sie die Komponente dem GridBagLayout-Manager hinzu fügen.

4 Layout-Manager-Wettbewerb (10 Zusatzpunkte)

Da der GridBagLayout-Manager in seiner Verwendung nervt, wäre ein Layout-Manager so ähnlich wie TableLayout, aber mit deutlich mehr Funktionalität doch viel praktischer zu verwenden.

Implementieren Sie ein erweitertes TableLayout, bei dem man alle Einstellungen von GridBadLayout global, für Zeilen und Spalten und für Zellen machen kann. Lokalere Einstellungen haben Vorrang vor globaleren (d. h. Zelle vor Zeile/Spalte etc.). Außerdem soll man Komponenten auch über mehrere Zellen positionieren können. Freie Tabellen-Felder sollen ebenfalls möglich sein.

Die besten Layout-Manager (in Bezug auf Funktionalität und Design) werden auf der Vorlesungshomepage präsentiert. Außerdem winken bis zu 10 attraktive Zusatzpunkte.

Viel Erfolg!