

Die Anforderung für Termin 1 besteht aus zwei Aufgaben:
Bereiten Sie sich gut vor, Sie werden am 17.10.2005 in der Vorlesung aufgefordert, Ihre Fragen dazu zu stellen. Nutzen Sie auch diese Gelegenheit.

Aufgabe 1:

Lernziele: java-Quelltext erstellen, compilieren, Anwendung starten. Import der benötigter Bibliotheken. Übergabeparameter an main() interpretieren. Verkettung von Strings. Abgeleitete Klasse erzeugen. Konstruktor und überladener Konstruktor. Layout-Manager: FlowLayout. Objekte und Felder von Objekten, dynamische Erzeugung von Feldern. for-Schleifen, if-else-Konstrukte

a) Schreiben Sie eine Fenster-Klasse, die von JFrame abgeleitet ist.
Schreiben Sie eine main-Funktion, welche dieses Fenster erzeugt, eine Größe setzt und anzeigt.

Welche Bibliothek(en) müssen Sie importieren?
Funktioniert das Schließen des Fensters?

b) Erweitern Sie die Anwendung folgendermaßen:

Wenn keine Parameter beim Aufruf angegeben werden, erscheint ein kurzer Hilfetext auf der Konsole + das Fenster ohne Labels.

Besorgen Sie sich den Container von JFrame und setzen Sie sein Layout auf FlowLayout.
Wenn Sie genau zwei Parameter angeben z.B. 4 Hallo werden 4 Labels mit dem Text „Hallo 1“, ..., „Hallo 4“ erzeugt und im Fenster angezeigt. Realisieren Sie dieses zu a)
unterschiedliche Verhalten über einen überladenen Konstruktor mit geeigneten Parametern.
Damit Sie mit der Anzahl Labels (JLabel) flexibel sind, realisieren Sie diese über ein Feld. Im Konstruktor wird dann die benötigte Anzahl erzeugt. Fügen Sie jedes einzelne Label dem Container hinzu.

c) Sobald die Anwendung funktioniert, rufen Sie diese mit einer größeren Anzahl von länderen Labels auf. Danach verändern Sie die Größe des Fensters und beobachten, was passiert.

Schreiben Sie die Beobachtung kurz in ein Protokoll in Ihre Dokumentmappe zum Praktikum.
Was folgern Sie aus diesen Beobachtungen für das Verhalten von FlowLayout

Nur, wenn genug Zeit oder freiwillige Zusatzaufgabe oder für zu Hause:

d) schreiben Sie eine Methode: `public Color getRandomColor()`, welche mit dem Zufallsgenerator `Math`. eine Farbe aus RGB erzeugt. Setzen Sie mit diesen Farben für jedes Label die Vorder- und den Hintergrundfarbe aus zufälligen Werten.

Warum sehen Sie möglicherweise die gesetzte Hintergrundfarbe noch nicht?

(Tip: was macht die Methode (der Super-Klasse JComponent)

`setOpaque(boolean isOpaque)?`)

Welche Kombinationen sind subjektiv gut, welche eher schlecht?

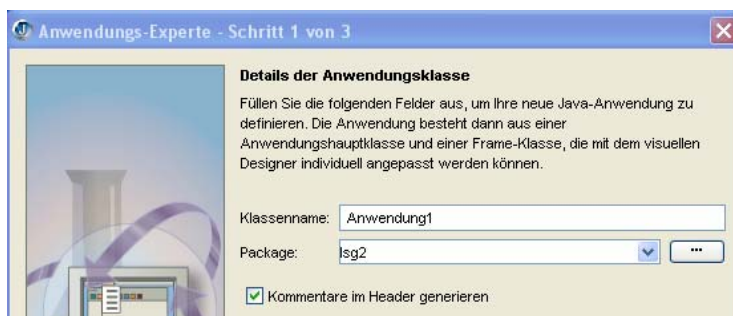
Aufgabe 2:

Lernziele: Vertrautmachung mit Borland JBuilder® Foundation, Erzeugung einer Anwendung, Einfügen grafischer Komponenten; BorderLayout

a) Erzeugen Sie mit dem JBuilder eine neue Anwendung.

Hierzu gehen Sie auf *Datei/Neues Projekt*. Geben Sie einen vernünftigen Namen an wie Termin1Aufg2. Setzen Sie einen sinnvollen Pfad. Lassen Sie die weiteren Einstellungen unverändert und folgen Sie mit *Weiter*> bis zur Fertigstellung.

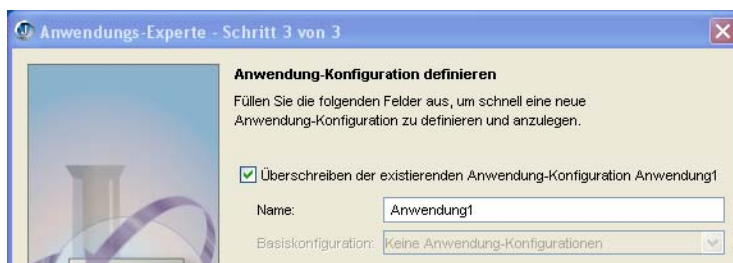
b) Mit Datei/Neu erzeugen Sie eine Anwendung. Folgen Sie den Schritten, siehe Abbildungen.



Kommentare im Header sind wünschenswert, also auswählen.



Derzeit keine weiteren Optionen aktivieren.

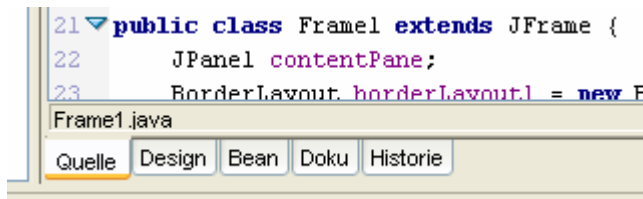


Setzen Sie die Check-Box zur Anwendungskonfiguration.

c) Compilieren Sie und führen Sie das Programm mit  aus.

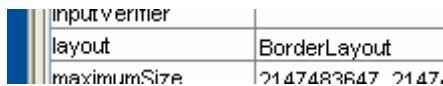
d) Lesen Sie den generierten Code Zeile für Zeile. Sie sollten ihn mit Ihren Kenntnissen bereits weitestgehend verstehen.

e) Wählen Sie jetzt in der Baumstruktur links die Dialogklasse aus, welche von `JFrame` abgeleitet ist. Benutzen Sie den Designer (*Design*), um Komponenten hinzuzufügen:



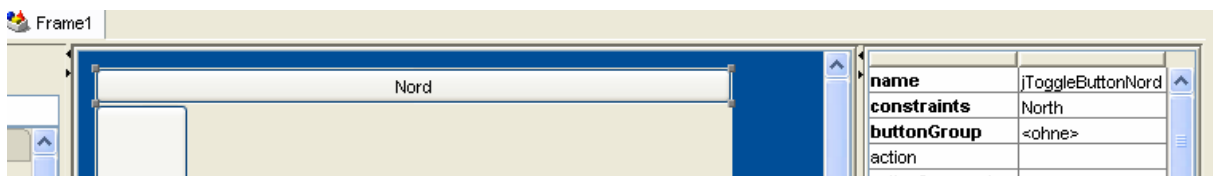
```
21 public class Frame1 extends JFrame {
22     JPanel contentPane;
23     BorderLayout borderLayout1 = new B
Frame1.java
Quelle Design Bean Doku Historie
```

d) Wählen Sie mit der Maus den *contentPane* aus. Den `BorderLayout` auswählen.



| | |
|--------------|------------------|
| input vermer | |
| layout | BorderLayout |
| maximumSize | 2147483647 2147. |

e) Erzeugen Sie nacheinander 5 `JToggleButton`, denen Sie die Namen und den Text „Nord“, „Sued“, „Ost“, „West“ und „Mitte“ geben. Plazieren Sie diese richtig im `contentPane`. Die constraints sollten dann die jeweils richtige Position „West“, „East“, „North“, „South“ und „Center“ anzeigen.



f) Compilieren und starten Sie die Anwendung und spielen Sie damit.

g) Betrachten Sie die Ergänzungen im generierten Code und versuchen Sie diesen zu verstehen.

h) Beschreiben Sie kurz im Protokoll, was `BorderLayout` bewirkt.

Führen Sie Ihre Lösungen jetzt vor, stellen Sie Fragen und lassen Sie sich das Praktikum testen.