

## Kontrollfragen A

1. Benennen Sie je 2 elementare GUI-Komponenten, Container und Layout-Manager.

GUI-Komponente:

*Button, List, Checkbox, Choice, Label, Scrollbar, TextField, TextArea*

Container:

*Panel, Frame*

Layout-Manager:

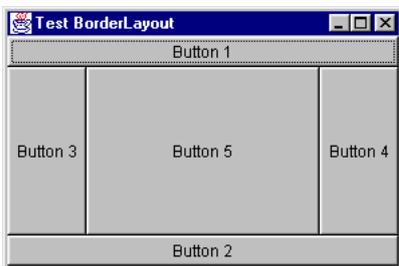
*BorderLayout, FlowLayout, GridLayout, setLayout (null)*

2. Wozu dient ein Panel?

*„Container“ um die einzelnen GUI-Komponenten zusammenzufassen.*

3. Illustrieren Sie, wie ein Frame seine GUI-Komponenten anordnet.

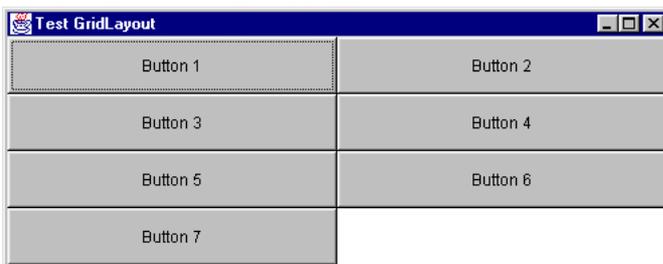
*setLayout(new BorderLayout());*



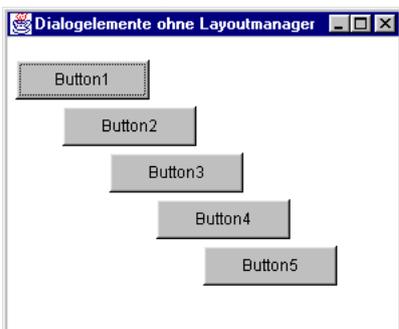
*setLayout(new FlowLayout());*



*setLayout(new GridLayout(4, 2));*



*setLayout(null);*



4. GUI-Komponenten besitzen unter anderem die Methoden `paint()` und `repaint()`. Wer ruft diese auf? Wann?

`paint()`

*Wer: System*

*Wann: Sobald sich etwas ändert im GUI, sowie wenn das Fenster wieder sichtbar wird*

`repaint()`

*Wer: Programm*

*Wann: im Code definiert*

*(repaint()) ruft die Methode paint() auf, resp. triggert das System dazu)*

5. `paint()` besitzt einen formalen Parameter `Graphics g`. Was repräsentiert `g`? Zählen Sie 3 Methoden auf, welche man für `g` aufrufen kann.

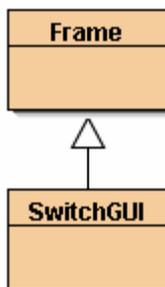
*Das „Zeichenblatt“, das Ausgabegerät z.B. Bildschirm, Drucker, etc.*

- `drawLine(...)`
- `drawOval(...)`
- `drawRect(...)`
- `drawString(...)`
- `fillOval(...)`
- `fillRect(...)`
- *etc.*

6. Wozu dient die Methode `add()` in den Beispielen 1 und 2?

*GUI-Komponenten dem Container hinzufügen*

7. Zeichnen Sie das Klassendiagramm zum Beispiel 1 auf.



## Kontrollfragen B

1. Was ist charakteristisch bei einem ereignisgesteuerten Programm?

- Im Normalfall "schläft" das Programm.
- Erst wenn ein Ereignis (Event) eintrifft, geschieht etwas.
- Was geschieht/ausgeführt wird, hängt vom Event ab.
- Nach der Ausführung fällt das Programm erneut in den "Schlafzustand".

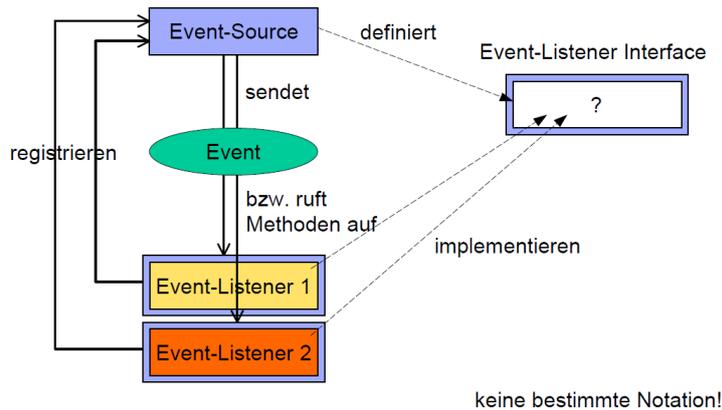
2. Beim Event-Handling ist das Zusammenspiel von Event, Event-Quelle und Event-Listener wichtig. Beschreiben Sie.

*Beispiel:*

*Event-Quelle: Button*

*Event: es wurde auf den Button geklickt*

*Event-Listener: Methode welche beim auslösen des Events ausgeführt wird.*



3. Welche Events erzeugen Button-Objekte?

*actionPerformed()*

4. Welche Events erzeugen Scrollbar-Objekte?

*adjustmentValueChanged()*

5. Listener kann man als eigenständige Klassen oder als innere Klassen (siehe SW5) implementieren. Welche Möglichkeit gibt es zudem?

*In einfachen Fällen können GUI-Klassen auch die ListenerInterfaces direkt selber implementieren bzw. zusätzlich auch die Rolle der Listeners wahrnehmen.*

6. Wozu dient die Methode addActionListener() (siehe Beispiele)? Bei wem ruft man diese Methode auf?

*Die Methode wird verwendet um einen „Action Listener“ bei einer „Action Source“ (z.B. Button) zu registrieren. Sie wird bei der GUI-Komponente aufgerufen (Button)*

```
Button on = new Button("On");
[...]
ButtonListener b1 = new ButtonListener(this);
on.addActionListener(b1);
```