Programmieren 2
Selbststudium Semesterwoche 4

Kapitel 11.1, 11.2 und 11.3
1. Das Konzept "Interface" ist wichtig und ist uns schon mehrfach begegnet. In welchen Zusammenhängen hatten wir es schon mit "Interfaces" zu tun?
➔ Simulierte Mehrfach-Vererbung (multiple inheritance)
➔ Explizite Design-Vorgabe
http://en.wikipedia.org/wiki/Interface_(Java)

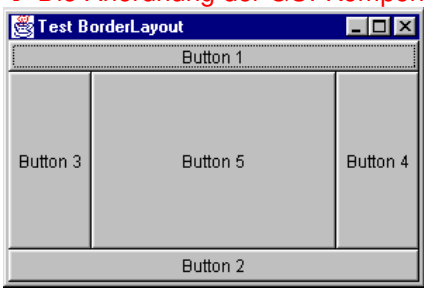
2. Erklären Sie mit Ihren eigenen Worten kurz die Konzepte GUIKomponente, GUILayout und EventHandling.
➔ GUI-Komponenten sind einzelne Funktionsträger (Bauteile), welche in einer Software als grafische Ein-/Ausgabehilfen eingesetzt werden können. Beispiele für GUI-Komponenten: Button, Textfeld, Textlabel, Dropdown- bzw. Combo-Box, Radio-Button, Check-Box, aber auch die Menüliste, Toolbar, uvm.
➔ Ein GUI-Layout (oder "Layout-Manager") erlaubt die Gestaltung eines Containers. Mit Hilfe von GUI-Layouts lässt sich somit die Ausrichtung, Füllfarbe, Ränder, Positionierung von Frames und Panels bestimmen.
➔ Event Handling: Ein Ereignis (engl. "Event") dient in der Softwaretechnik zur Steuerung des Programmflusses. Das Programm wird nicht linear durchlaufen, sondern es werden spezielle Ereignisbehandlungsroutinen (engl. listener, observer, event handler) immer dann ausgeführt, wenn ein bestimmtes Ereignis auftritt (vergleiche Rückruffunktion). Ein verwandtes Konzept sind Interrupts. Ereignisse eignen sich besonders gut zur Implementierung von grafischen Benutzeroberflächen, wobei hier die Ereignisse meist Aktionen des Benutzers sind, wie zum Beispiel das Drücken einer Taste oder das Anklicken einer Schaltfläche.

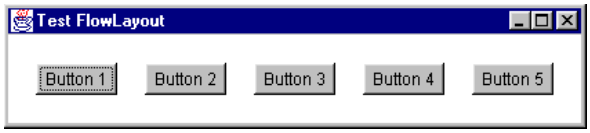
3. Inwiefern hängen AWT und Swing zusammen?
➔ Swing gehört zu den Java Foundation Classes (JFC), die eine Sammlung von Bibliotheken zur Programmierung von grafischen Benutzerschnittstellen bereitstellen. Zu diesen Bibliotheken gehören Java2D, das Accessibility-API, das Drag & Drop-API und das Abstract Window Toolkit (AWT). Swing baut auf dem älteren AWT auf und ist mit den anderen APIs verwoben.

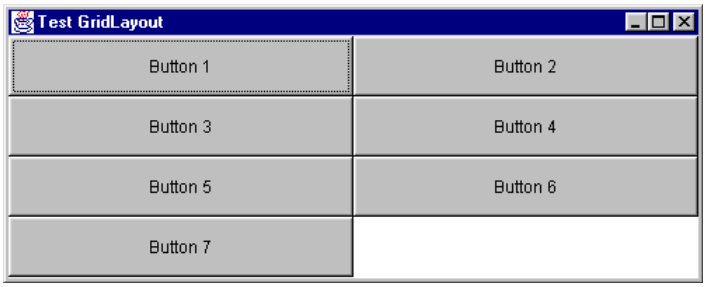
Handout PRG2_SW4_OOP
4. Gehen Sie nochmals die Kontrollfragen A und B durch:
4.1. Benennen Sie je 2 elementare GUIKomponenten, Container und LayoutManager.
➔ Elementare GUI-Komponenten: Button, TextField, Label,…
➔ Container: Frame, Panel
➔ LayoutManager: BorderLayout, FlowLayout, GridLayout, Null-Layout,…

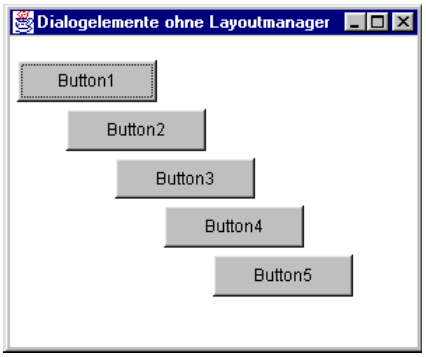
4.2. Wozu dient ein Panel?
➔ Panel ist die einfachste Container Klasse. Eine Anwendung kann Panels verwenden, um andere Komponenten (und/oder Container) zu verschachteln.
http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Panel.html

4.3. Illustrieren Sie, wie ein Frame seine GUI-Komponenten anordnet.
➔ Die Anordnung der GUI-Komponenten in einem Frame ist abhängig vom zugewiesenen LayoutManager.









4.4. GUI-Komponenten besitzen unter anderem die Methoden paint() und repaint().
Wer ruft diese auf? Wann?
➔ Die Methode paint() wird vom Betriebssystem aufgerufen, sobald z.B. die Grösse des Fensters geändert wird.
➔ Die Methode repaint() kann innerhalb der Java Anwendung aufgerufen werden und erzwingt indirekt den Aufruf der paint() Methode.

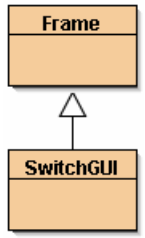
4.5. paint() besitzt einen formalen Parameter Graphics g. Was repräsentiert g? Zählen Sie 3 Methoden auf, welche man für g aufrufen kann.
➔ Graphics ist eine abstrakte Basisklasse für alle Grafikanwendungen.
➔ Graphics stellt grundlegende Zeichenfunktionen zur Verfügung wie z.B.
- drawLine(…)
- drawOval(…)
- drawRect(…)
- drawString(…)
- fillOval(…)
- fillRect(…)

➔ Graphics speichert auch Status Informationen für die gezeichneten Objekte:
- den Component, auf welchem die Zeichenanwendung statt finden soll
- die gegenwärtige Farbe
- die gegenwärtige Schriftart

http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Graphics.html

4.6. Wozu dient die Methode add() in den Beispielen 1 und 2?
➔ Da die Klasse SwitchGUI die Klasse Frame erweitert, wird die Methode add(…) von der Klasse Frame geerbt. Mit dieser Methode werden Komponenten dem Container (Frame) hinzugefügt.

4.7. Zeichnen Sie das Klassendiagramm zum Beispiel 1 auf.



5. Implementieren Sie das Beispiel 1 (SwitchGUI). Erweitern Sie das GUI so, dass man auch über einen Button "Close" an der Position: BorderLayout.EAST die Anwendung beenden kann.

```
import java.awt.*;
import java.awt.event.*;

public class SwitchGUI extends Frame implements ActionListener
{
    // GUI Elemente erzeugen (Zugriffsmodifizierer ist je "Package"!)
    Label state = new Label("The Switch is off.");
    Button on = new Button("On");
    Button off = new Button("Off");
    Button close = new Button("Close");

    public SwitchGUI()
    {
        // Frame initialisieren
        setTitle("Switch");
        setSize(150, 150);
        setResizable(false);

        // LayoutManager setzen
        setLayout(new BorderLayout());

        // ActionListener des Frames für close Button setzen
        close.addActionListener(this);

        // GUI zusammensetzen
        add(on, BorderLayout.NORTH);
        add(state, BorderLayout.CENTER);
        add(off, BorderLayout.SOUTH);
        add(close, BorderLayout.EAST);

        // Frame sichtbar machen
        setVisible(true);
    }

    public void windowIconified(WindowEvent e) { }
    public void windowOpened(WindowEvent e) { }
    public void windowClosed(WindowEvent e) { }
    public void windowDeiconified(WindowEvent e) { }
    public void windowActivated(WindowEvent e) { }
    public void windowDeactivated(WindowEvent e) { }

    public void actionPerformed(ActionEvent e)
    {
        System.exit(0);
    }
}
```

6. Implementieren Sie in Anlehnung an das Beispiel 2 (LineGUI) eine neue Anwendung CircleGUI. Über eine Scrollbar lässt sich der Radius eines Kreises vorgeben, welcher entsprechend im Frame gezeichnet wird.


```
import java.awt.*;
import java.awt.event.*;

public class CircleGUI extends Frame implements ActionListener, WindowListener,
AdjustmentListener
{
    // GUI Elemente erzeugen
    private int diameter;
    private Scrollbar slider;

    //(orientation, value, size of the bubble, minimum, maximum)

    public CircleGUI()
    {
        // Frame initialisieren
        setTitle("CircleGUI");
        setSize(200, 200);
        setResizable(false);

        // Scrollbar initialisieren
        slider = new Scrollbar(Scrollbar.HORIZONTAL, 0, 1, 0, 100);
        slider.addAdjustmentListener(this);

        // LayoutManager setzen
        setLayout(new FlowLayout());

        // GUI zusammensetzen
        add(slider);

        // Implementiert WindowListener
        addWindowListener(this);

        // Frame sichtbar machen
        setVisible(true);
    }

    public CircleGUI(int diameter)
    {
        this();
        setDiameter(diameter);
    }

    public void setDiameter(int diameter)
    {
        this.diameter = diameter;
        repaint();
    }

    // Direkt in das Frame zeichnen
    // Methode paint() von Frame wird überschrieben
```

```
    public void paint(Graphics g)
    {
        super.paint(g); // paint() der Oberklasse aufrufen
        this.diameter = slider.getValue();
        g.fillOval(100-(diameter/2), 120-(diameter/2), diameter, diameter);
    }

    // WindowListener
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }

    // ActionListener
    public void windowIconified(WindowEvent e) { }
    public void windowOpened(WindowEvent e) { }
    public void windowClosed(WindowEvent e) { }
    public void windowDeiconified(WindowEvent e) { }
    public void windowActivated(WindowEvent e) { }
    public void windowDeactivated(WindowEvent e) { }
    public void actionPerformed(ActionEvent e) { }

    // AdjustmentListener
    public void adjustmentValueChanged(AdjustmentEvent e)
    {
        setDiameter(e.getValue());
    }
}
```

http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Scrollbar.html
http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Graphics.html
http://www.mpi-inf.mpg.de/departments/d5/teaching/ss05/is05/java/GoToJava2.2/html/k100211.html

Handout PRG2_SW4_DAT
7. Gehen Sie nochmals die Kontrollfragen A, B und C durch.
8. Implementieren Sie die Klasse Balloon gemäss Handout. Zusätzlich soll ein Ballon aber das Attribut Color color (vgl. java.awt.Color) besitzen. Zwei Ballone sind genau dann gleich, wenn Sie in der Aufschrift (text) und in der Farbe (color) übereinstimmen. Passen Sie equals() entsprechend an. Testen Sie equals() mit Hilfe der Methode main() oder mit JUnit.
9. Welche Methode müssen Sie nun auch noch anpassen?

```
import java.util.Set;
import java.util.HashSet;
import java.awt.Color;

public class Balloon
{
    private String text;
    private int size;
    private Color color;

    public Balloon(String t, Color c)
    {
        text = t;
        color = c;
        size = 0;
    }

    public void blowUp()
    {
        size = size + 5;
    }

    public void deflate()
    {
        size = 0;
    }
```

```
    public boolean equals(Object other)
    {
        // 1. Test auf Identität
        if (this == other)
            return true;

        // 2. Test auf null
        if (other == null)
            return false;

        // 3. Test auf Vergleichbarkeit
        if (other.getClass() != this.getClass())
            return false;

        // 4. Vergleich relevanter Felder
        if ((!text.equals(((Balloon)other).text) || (!color.equals(((Balloon)other).color
))))
            return false;
            return true;
     }

     public static void main(String[] args)
     {
         Balloon b1 = new Balloon("Hochschule Luzern", Color.BLACK);
         Balloon b2 = new Balloon("Hochschule Luzern", new Color(0,0,0));
         Balloon b3 = new Balloon("Hochschule Luzern", Color.WHITE);
         Balloon b4 = new Balloon("Modul Programmieren 2", Color.BLACK);

         b2.blowUp();
         b3.blowUp();

         System.out.println(b1.equals(b1));
         System.out.println(b1.equals(b2));
         System.out.println(b1.equals(b3));
         System.out.println(b1.equals(b4));

         System.out.println(b2.equals(b2));
         System.out.println(b2.equals(b3));
         System.out.println(b2.equals(b4));

         System.out.println(b3.equals(b3));
         System.out.println(b3.equals(b4));

         System.out.println(b4.equals(b4));
     }
}
```

Vergleichsbefehl	Vergleich von "text":	Vergleich von "text" und "color":
System.out.println(b1.equals(b1));	true	true
System.out.println(b1.equals(b2));	true	true
System.out.println(b1.equals(b3));	true	false
System.out.println(b1.equals(b4));	false	false
System.out.println(b2.equals(b2));	true	true
System.out.println(b2.equals(b3));	true	false
System.out.println(b2.equals(b4));	false	false
System.out.println(b3.equals(b3));	true	true
System.out.println(b3.equals(b4));	false	false
System.out.println(b4.equals(b4));	true	true