

Übung 2: *LandscapePanel*

Motivation

In dieser Übung lernen Sie, wie verschiedene Methoden der Klasse **Graphics** zum Zeichnen einer einfachen Grafik in einem **Panel** verwendet werden können.

Aufgabe 1: Grundgerüst

- Kopieren den Ordner **LandscapePanel** in den **Workspace** und fügen Sie es als Projekt dem **Workbench** zu. Die Übung enthält in der Datei **LandscapeFrame.java** im wesentlichen nachfolgendes Grundgerüst:

```
class LandscapePanel extends JPanel {  
  
    public void init() {  
    }  
  
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);  
    }  
    // ...  
}
```

- Das zur Darstellung notwendige Frame am Ende der Datei, blenden wir im Moment aus unserem Denken aus.

Aufgabe 2: Let's draw

Ergänzen Sie den Code so, dass in etwa die nachfolgende Zeichnung entsteht:

Es stehen die folgenden Methoden zur Verfügung:

drawLine (x1, y1, x2, y2)

x1, y1: Koordinate des Linienanfangs
x2, y2: Koordinate des Linienendes

drawRect (x, y, width, height)

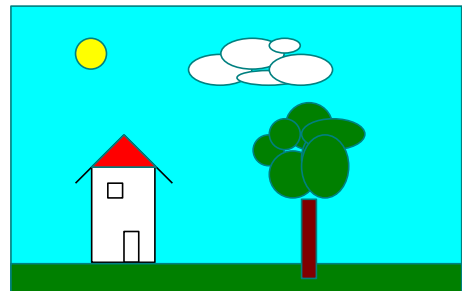
x1, y1: Koordinate der **linken** oberen Ecke
width, Height: Breite, bzw. Höhe des Rechtecks

drawOval (x, y, width, height)

Die Parameter beschreiben hier ein Rechteck, welches die Ellipse umgibt.

drawArc (x, y, width, height, a1, a2)

x1, y1: Koordinate der **linken** oberen Ecke
a1, a2: Anfangswinkel und Winkel des zu zeichnenden Bogens.



Die folgenden Farben können als Zeichen- oder Hintergrundfarbe festgelegt werden:

black	blue	Cyan	darkGray	gray
green	lightGray	magenta	orange	pink
red	white	Yellow		

Mit **setBackground(Color.lightGray)** kann die Hintergrundfarbe gesetzt werden, analog dazu wird mit **g.setColor(Color.red)** die Zeichenfarbe festgelegt.

In Java stehen auch Methoden zum Füllen von Formen sowie weitere Zeichen-Methoden zur Verfügung:

fillArc(), fillRect(), fillOval()

ihre Parameter sind identisch mit denen der draw-Methoden.

Aufgabe 3: Animation

Ergänzen Sie den Code wie folgt, so dass eine Animation entsteht. In der Animation sollen sich die Wolken über die Landschaft hinweg bewegen. Als Beispiel wird im folgenden Code ein Rechteck der Groesse 20X20 über den Bildschirm verschoben.

```
public void init() {  
    setBackground(new Color(145, 245, 245));  
    new SimpleTimer(50, new SimpleTimerListener() {  
        public void timerAction() {  
            repaint();  
        }  
    }).start();  
}
```

Unser Programm verwendet einen `SimpleTimer` in der Art, dass das Bild in regelmässigen Abständen wieder neu gezeichnet wird. Dies geschieht mittels Aufruf von `repaint()` – `paintComponent()` .