

Generics

- Schablonen für Klassen, Interfaces oder Methoden, die noch nicht festgelegte Datentypen verwenden
- *Typvariablen* statt Datentypen
 - werden hinter Bezeichnern in spitzen Klammern aufgezählt
 - Variablen werden im folgenden Namensraum wie Datentypen verwendet

```
class Pair<T,U> { private T first;  
                private U second;  
                Pair(T elem_1, U elem_2) { ... } }
```

- in der Anwendung: Angabe eines konkreten Datentyps für jede Typvariable
 ↪ *generischer Typ*

```
Pair<String, Integer> psi;
```

Graphische Benutzeroberflächen

- GUI: Graphical User Interfaces
- bestehen aus Menüs, Fenstern, Eingabefeldern, Buttons usw.
- Ein Paket, das dafür notwendige Klassen enthält , ist das *Abstract Windowing Toolkit* (`java.awt`).
- AWT stellt darüber hinaus Klassen zum Erzeugen von Graphiken zur Verfügung.

Einige GUI-Elemente in AWT

Frame	Fenster mit Rahmen
Button	Pushbutton
Label	einzeiliger Text
Checkbox	Markierungsfeld
Choice	PopUp-Auswahlfelder
Scrollbar	Schiebebalken

Alle diese Elemente sind von der abstrakten Klasse `java.awt.Component` abgeleitet und werden daher *Komponenten* genannt.

Container sind Komponenten, die (im Gegensatz zu einfachen Komponenten wie Labels oder Buttons) andere Komponenten aufnehmen können.

Methoden von Containern

- `setForeground(Color c)` (Vordergrundfarbe)
- `setBackground(Color c)` (Hintergrundfarbe)
- `setFont(Font font)` (Schriftart und -größe)
- `setSize(int x, int y)` (Größe der Komponente)
- `setVisible(true)`, `setVisible(false)` (Sichtbarkeit)
- `validate()` (Aktualisierung der Bildschirmdarstellung)
- `add(Component c)` (Hinzufügen der Komponente c)
- `paint(Graphics g) {}` (Zeichnen des Containers im Graphikkontext g)

```
public void paint (Graphics g) {  
    g.drawString("Hallo", 50, 50);  
}
```

Wichtige Methoden der Graphics-Klasse

`drawLine(x1,y1,x2,y2)`

Linie von (x1,y1) nach (x2,y2)

`drawRect(x,y,width,height)`

Rechteck mit linker oberer
Ecke in (x,y)

`fillRect(x,y,width,height)`

gefülltes Rechteck

`drawOval(x,y,width,height)`

Ellipse (im Rechteck der
angegebenen Parameter)

`fillOval(x,y,width,height)`

gefüllte Ellipse

`drawArc(x,y,width,height,start,ende)`

Kreisbogen von start bis ende
mit Werten von 0 ("Osten")
über 90 ("Norden") usw.

`fillArc(x,y,width,height,start,ende)`

gefüllter Kreisbogen

Graphics-Methoden (2)

<code>drawPolygon(x[],y[],n)</code>	Polygonenzug durch n Punkte
<code>fillPolygon(x[],y[],n)</code>	gefülltes Vieleck
<code>drawString(str,x,y)</code>	Ausgabe eines Strings
<code>setColor(Color c)</code>	setzt die Vordergrundfarbe auf ein Datenelement der Klasse Color, z.B. auf blue, green, red, yellow, white, gray, black
<code>setFont(Font font)</code>	setzt den Schriftsatz auf ein Objekt der Klasse Font

Objekte der Font-Klasse

Der Konstruktor erhält drei Parameter:

1. die Schriftart, z.B. `Serif`, `SansSerif`, `Monospaced`,
2. eine Darstellungsart, die sich aus der Summe von `BOLD`, `ITALIC`, `PLAIN` ergeben kann,
3. die Schriftgröße in `pt`

Beispiel:

```
Font font = new Font("Serif", Font.BOLD+Font.ITALIC, 24);
```