

Generics

- Schablonen für Klassen, Interfaces oder Methoden, die noch nicht festgelegte Datentypen verwenden
- *Typvariablen* statt (fester) 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 (1)

- 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.
- Grundlage CUA-Standard

Graphische Benutzeroberflächen (2)

Der Common User Access (CUA)-Standard

Wichtige Etappen und Standardwerke hierzu sind:

1. CUA Interface Guide, IBM 1989, 1991,
2. Windows Interface Guidelines for Software Design, Microsoft 1995,
3. OSF/Motif Style Guide, Open Software Foundation 1992.

Der CUA-Standard

- definiert das Aussehen, die Anordnung und die Benutzung grafischer Benutzeroberflächen.
- Er dient der Vereinheitlichung der Applikationen, so dass jeder Benutzer intuitiv jede Applikation bedienen kann.
- Es werden dabei sowohl das Aussehen als auch der Name jedes sichtbaren Gebildes definiert.

Graphische Benutzeroberflächen (3)

Ausgewählte Elemente des CUA-Standards:

Name	Beschreibung
button	die grafische Repräsentation eines choice
choice	etwas, was der Nutzer anwählen kann
check box	eine rechteckige Box, in welcher die Auswahl markiert wird
cursor	die Anzeige der Position für Keyboard Aktionen
dialog	ein Interaktionsfenster zwischen Benutzer und Computer
group	logische Zusammenfassung von Objekten
list box	eine Auswahlliste von Objekten

Graphische Benutzeroberflächen (3)

Ausgewählte Elemente des CUA-Standards:

Name	Beschreibung
menu	eine Auswahlliste für choices
pointer	die Anzeige der Position für Mausaktionen
Title bar	Titelzeile des Fensters

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

<code>drawLine(x1,y1,x2,y2)</code>	Linie von (x1,y1) nach (x2,y2)
<code>drawRect(x,y,width,height)</code>	Rechteck mit linker oberer Ecke in (x,y)
<code>fillRect(x,y,width,height)</code>	gefülltes Rechteck
<code>drawOval(x,y,width,height)</code>	Ellipse (im Rechteck der angegebenen Parameter)
<code>fillOval(x,y,width,height)</code>	gefüllte Ellipse
<code>drawArc(x,y,width,height,start,ende)</code>	Kreisbogen von start bis ende mit Werten von 0 ("Osten") über 90 ("Norden") usw.
<code>fillArc(x,y,width,height,start,ende)</code>	gefüllter Kreisbogen

Graphics-Methoden (2)

`drawPolygon(x[],y[],n)`

Polygonenzug durch n Punkte

`fillPolygon(x[],y[],n)`

gefülltes Vieleck

`drawString(str,x,y)`

Ausgabe eines Strings

`setColor(Color c)`

setzt die Vordergrundfarbe

auf ein Datenelement der Klasse `Color`,
z.B. auf `blue`, `green`, `red`, `yellow`,
`white`, `gray`, `black`

`setFont(Font font)`

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);
```