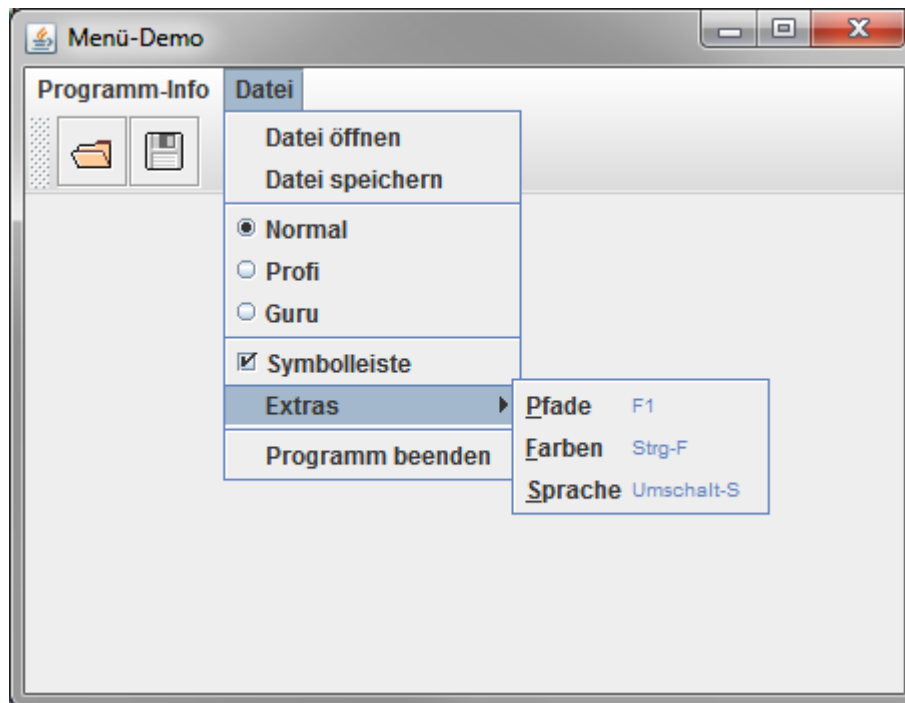


MENÜS

htw.


Erstellen von Menüs

- 1. Menüelemente (**JMenuItem**) definieren und Ereignisbehandlung festlegen
- 2. Menüelemente einer Menüleiste (**JMenuBar**) hinzufügen und die Menüleiste dem Fenster hinzufügen



Grundgerüst Menü

(MenuDemo.java)



```
public class MenuDemo extends JFrame implements ActionListener {  
  
    public MenuDemo() {  
        super();  
  
        // Menüleiste  
        JMenuBar menuLeiste = erzeugeMenueLeiste();  
        setJMenuBar(menuLeiste);  
    }  
  
    // Methode für Menüleiste - siehe nächste Folie  
  
}
```

Grundgerüst Menü

(MenuDemo.java)

```
// Diese Methode stellt die Menüleiste zusammen
private JMenuBar erzeugeMenueLeiste()
{
    JMenuBar menueLeiste = new JMenuBar();

    // Menü Datei
    JMenu menuDatei = new JMenu("Datei");
    JMenuItem menuDateiOeffnen = new JMenuItem("Datei öffnen",
                                                new ImageIcon("Open.gif"));
    menuDateiOeffnen.addActionListener(this);
    JMenuItem menuDateiSpeichern = new JMenuItem("Datei speichern",
                                                  new ImageIcon("Save.gif"));
    menuDateiSpeichern.addActionListener(this);

    // das Menü Datei zusammenstellen
    menuDatei.add(menuDateiOeffnen);
    menuDatei.add(menuDateiSpeichern);

    // Menüleiste zusammenstellen
    menueLeiste.add(menuDatei);

    return menueLeiste;
}
```

Menü

Menü – die Klassen

- **JMenuBar:**
 - `JMenuBar()` : erstellt eine neue Menüleiste (mit `setJMenuBar(menuLeiste);`) an das Fenster binden
 - `add(JMenu m)` : hängt das Menü `m` an die Menüleiste an
- **JMenu:**
 - `JMenu(String s)` : erstellt ein neues Menü mit dem Text `s`
 - `add(JMenuItem)` : hängt einen Menüeintrag an das Menü
 - `add(Component c)` : `c` meistens `JMenu` → zum Anhängen von Untermenüs
 - `addSeparator()` : Einfügen einer Trennlinie
- **JMenuItem:**
 - `JMenuItem(String text)` : Menüeintrag „text“
 - `JMenuItem(String text, Icon icon)` : Menüeintrag „text“ mit Icon
 - `JMenuItem(String text, int mnemonic)` : Menüeintrag mit Tastenkürzel

Untermenüs

(MenuDemo.java)

```
JMenu menuDatei = new JMenu("Datei");

// Untermenü Datei/Extras
JMenu menuDateiExtras = new JMenu("Extras");
JMenuItem menuDateiExtrasPfade = new JMenuItem("Pfade", 'P');
menuDateiExtrasPfade.addActionListener(this);
JMenuItem menuDateiExtrasFarben = new JMenuItem("Farben", 'F');
menuDateiExtrasFarben.addActionListener(this);
JMenuItem menuDateiExtrasSprache = new JMenuItem("Sprache", 'S');
menuDateiExtrasSprache.addActionListener(this);

// das Untermenü Extras zusammenstellen
menuDateiExtras.add(menuDateiExtrasPfade);
menuDateiExtras.add(menuDateiExtrasFarben);
menuDateiExtras.add(menuDateiExtrasSprache);

// das Menü Datei zusammenstellen
menuDatei.add(menuDateiOeffnen);
menuDatei.add(menuDateiSpeichern);
menuDatei.add(menuDateiExtras);

menueLeiste.add(menuDatei);
```

- anstelle von JMenuItem wird ein JMenu hinzugefügt

Kontextmenüs (Popup-Menüs)

- unter Verwendung der Klasse JPopupMenu
 - JPopupMenu() : erzeugt ein Kontextmenü
 - add(String s) bzw. add(JMenuItem i) : fügt einen Eintrag hinzu
 - addSeparator() : fügt einen Trennstrich hinzu
 - show(Component c, int x, int y): zeigt das Kontextmenü in der Komponente c an Position (x,y) an
- durch Rechtsklick aufgerufen (Implementierung des Kontextmenüs in einer MouseEvent-Ereignisbehandlung, z.B. mousePressed (alle) und mouseReleased (Windows) → `if(e.isPopupTrigger()) erzeugeKontextmenue(e);`)
- siehe MenuDemo.java

Kontextmenüs

(MenuDemo.java)

```
// erzeugt ein Kontext-Menü und zeigt es an
private void erzeugeKontextMenue(MouseEvent e) {
    JPopupMenu popup = new JPopupMenu();

    JMenuItem popupKopieren = new JMenuItem("Kopieren");
    popupKopieren.addActionListener(fenster);

    JMenuItem popupEinfuegen = new JMenuItem("Einfügen");
    popupEinfuegen.addActionListener(fenster);

    JMenuItem popupEnde = new JMenuItem("Programm beenden");
    popupEnde.addActionListener(fenster);

    popup.add(popupKopieren);
    popup.add(popupEinfuegen);
    popup.addSeparator();
    popup.add(popupEnde);

    // anzeigen an aktueller Mausposition
    popup.show(e.getComponent(), e.getX(), e.getY());
}
```

- in *mouseReleased*: `erzeugeKontextMenue(e);`

Ereignisbehandlung

- Klick auf ein JMenuItem → ActionEvent (wie JButton)
- für JCheckBoxMenuItem → ItemEvent (itemStateChanged())
- Kontextmenü: MouseEvent (mouseReleased) ; für die Menüeinträge dann wieder ActionListener
- Beispiel: Erscheinen und Verschwinden der Symbolleiste (JToolBar);
siehe MenuDemo.java

```
public void itemStateChanged(ItemEvent e) {  
    Object item = e.getItemSelectable();  
  
    if(item == menuDateiCISymbolleiste) {  
        if(e.getStateChange() == ItemEvent.DESELECTED)  
            getContentPane().remove(symbolleiste);  
        else  
            getContentPane().add(symbolleiste, BorderLayout.NORTH);  
  
        repaint();  
    }  
}
```