

# Graphische Nutzerschnittstellen

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- [mwilhelm@hs-harz.de](mailto:mwilhelm@hs-harz.de)
- <http://www.miwilhelm.de>
- Raum 2.202
- Tel. 03943 / 659 338

# Inhalt

1. Einführung, Literatur, Begriffe
- 2. Architektur eines Fenstersystems**
3. JavaFX
4. Dialoge in JavaFX
5. Grafik in JavaFX
6. Benutzeroberfläche (Dialog, SDI, MDI, SDI, RDI)
7. Testroutinen (JUnit)
8. JDBC (Datenbankanbindung)

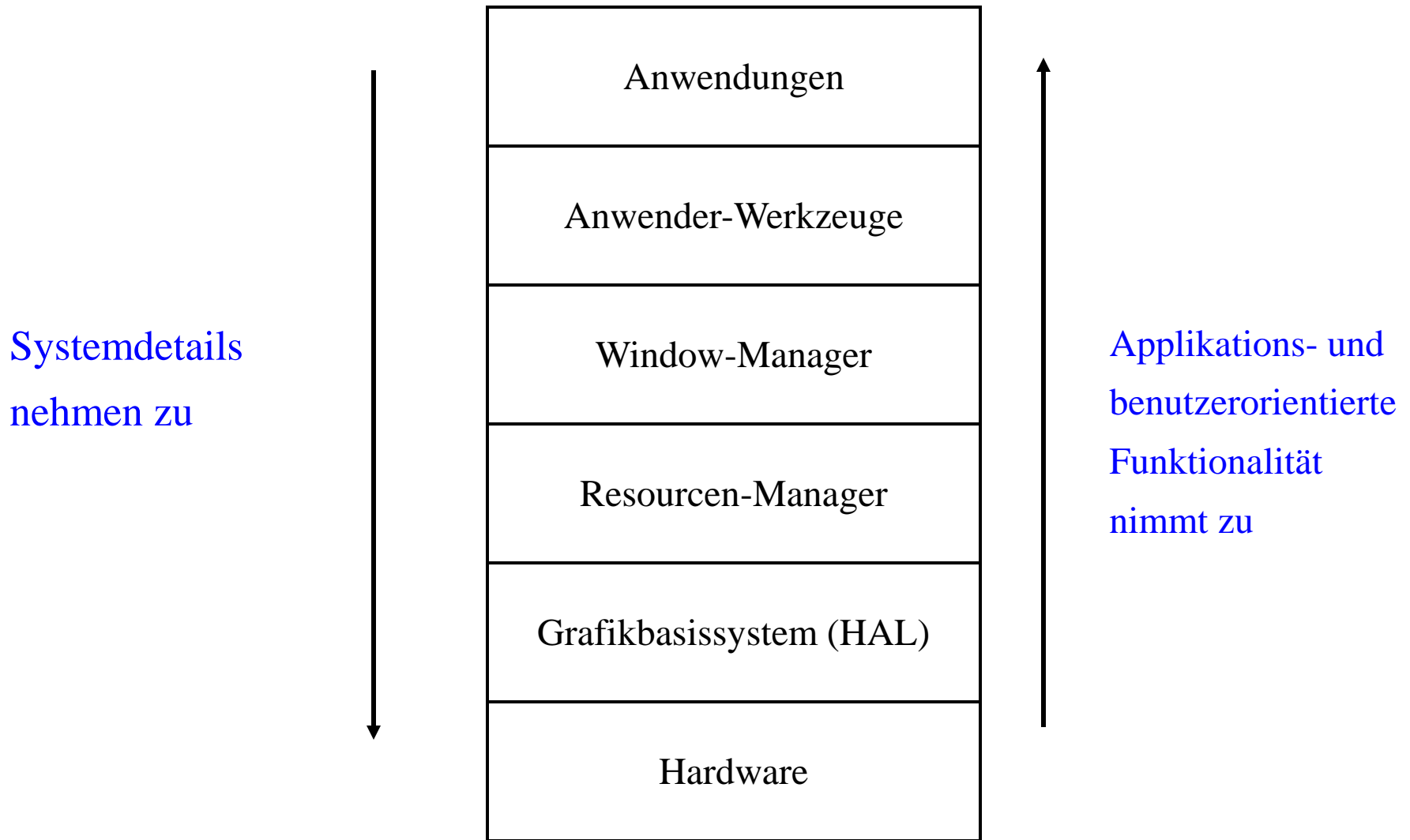
# Architektur eines Fenstersystems:

- **Kriterien zur Bewertung eines Fenstersystems**
- **Komponenten eines Fenstersystems**
- **Die Grafikbasis**
- **Ressourcenverwaltung**
- **Betriebssystem und Fenstersystem**
- **Fenstermontierer / Window Management**
- **Standard-Interaktionsobjekte**

# Kriterien zur Bewertung eines Fenstersystems

- Verfügbarkeit
- Vielfältige Hardware
- Grafikleistung
- einfache Programmierung
- Betriebsmittelverwaltung verbergen
- Parallelität (Threads)
  - extern: Parallele Funktionen für die Benutzer
  - intern: nebenläufige Module
- Grafikgrundmodell
  - Rastermodell: Swing, Delphi, MFC, WinForms
  - Vektormodell: Windows Presentation Foundation (WPF)
- Guide-Style
  - fest vorgegeben vs. variabel
- Erweiterbarkeit
  - nicht erweiterbar (MyJLabel)
  - Quellcodeänderung
  - dynamisch erweiterbar
- Anpassbarkeit
  - Benutzerspezifisch
  - Landesspezifisch
  - Lokalization (statisch, dynamisch)
- Teilbarkeit der Ressourcen
  - Gemeinsame Ausgabe in mehreren Applikationen (Rechte etc.)
  - Verteilte Fenstersysteme
- Struktur des API (OOP, Prozedural)
- Komfort des API (Prototyp)
- Unabhängigkeit von Applikation und Interaktion (GUI vs. Code)
- Kommunikation zwischen den Applikationen
  - Zwischenablage
  - DDE
  - OLE
  - (D)COM

# Komponenten eines Fenstersystems



# Komponenten eines Fenstersystems

## **Anwender-Werkzeuge**

Toolkit-Schicht. Definiert Standard-Dialogfenster

## **Window-Manager**

Verwaltet die Anzeige im Fenster, Resize etc.

## **Ressourcen-Manager**

Verwaltet die logischen Grafikelemente (Fonts, Stifte etc). Auch die gemeinsame Nutzung

## **Grafikbasissystem (HAL)**

Verwaltet die Zugriffe auf die verschiedenen Grafiksubsysteme (Screen, Printer, WMF)

## **Hardware**

# Die Grafikbasis

- Eingabeaktionen
  - Parameterrückbereitung der Grafikbefehle
- Zeichenmodus
  - Copy
  - And
  - Oder
  - Xor
- Elementare Operationen
  - Funktion BitBlt (Win API, Kopieren von Bildblöcken)
  - Integer-Arithmetik, schneller als Floating-Points (Problem der Normalisierung)
  - Clipping (Linie, Kreis, Rechteck, Text)
- Zeichnen von Texten
  - **int DrawText( const CString& str, LPRECT lpRect, UINT nFormat );**

# Objekte der Grafikbasis

- Zeichenflächen
- geräteunabhängig (Zwischenablage, Dateien)
- geräteabhängig (Drucker, Fax, Bildschirm)
- Ausgabe-Objekte
- elementar (direkt von der Hardware unterstützt, DirectX 13)
- zusammengesetzt (Zeichensätze)
- Grafischer Kontext (Attribute einer Zeichenfläche)
- Strichbreite, Strichart
- Schriftart
- Farbtabelle



# Standardinteraktionsobjekte (GUI-Elemente)

- Text (Label)
- Schreibmarke, Mauszeiger (Cursor)
- Schalter (mit, ohne Bitmap)
- Auswahl (Menü, Popupmenü, Untermenüs)
- Schalter (Anwahlschalter, Auswahlsschalter, exklusiver Auswahlsschalter, Gruppenschalter)
- Eingabefeld (Edit)
- Spinfeld
- Listenfeld (List)
- Eingabefeld mit Listenfeld (ComboBox)
- Rollbalken (Scrollbar)
- ProgressBar
- Register
- Fenster (Frame, Window)

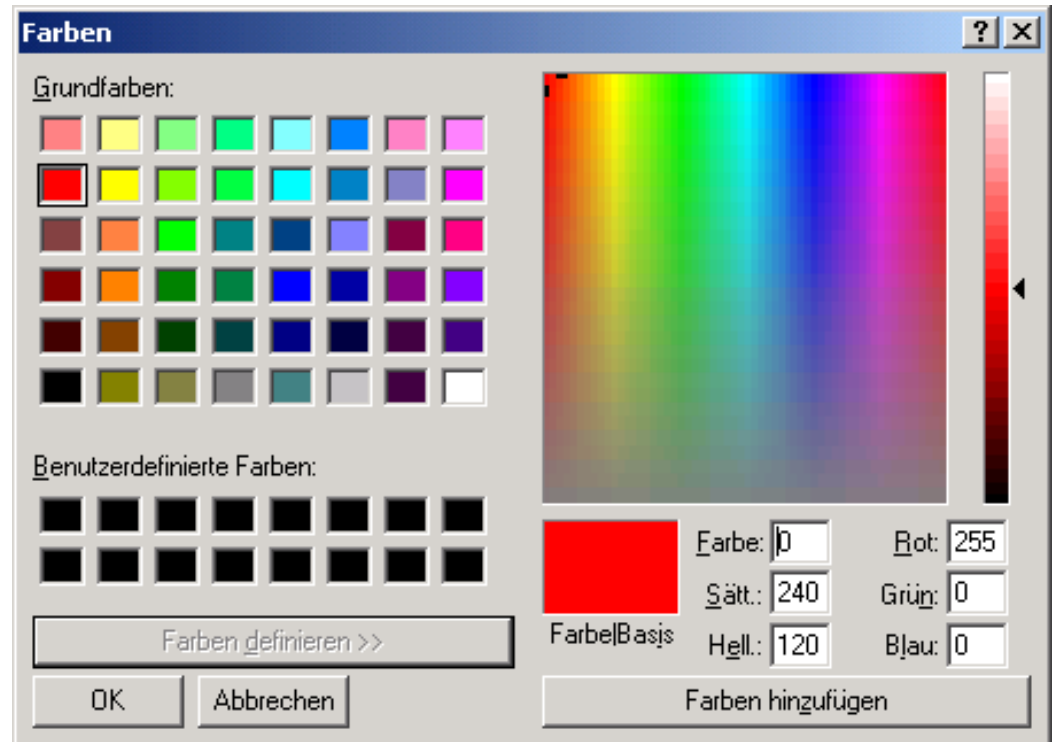
# Standardinteraktionsobjekte

## Dialogbausteine „Common dialogboxes“

- GetOpenFileName
- GetSaveFileName
- PrintDlg
- FindText / NextSearch
- ReplaceText
- ChooseColor
- ChooseFont
  
- Methoden:
  - GetDialogTitle
  - SetDialogTitle

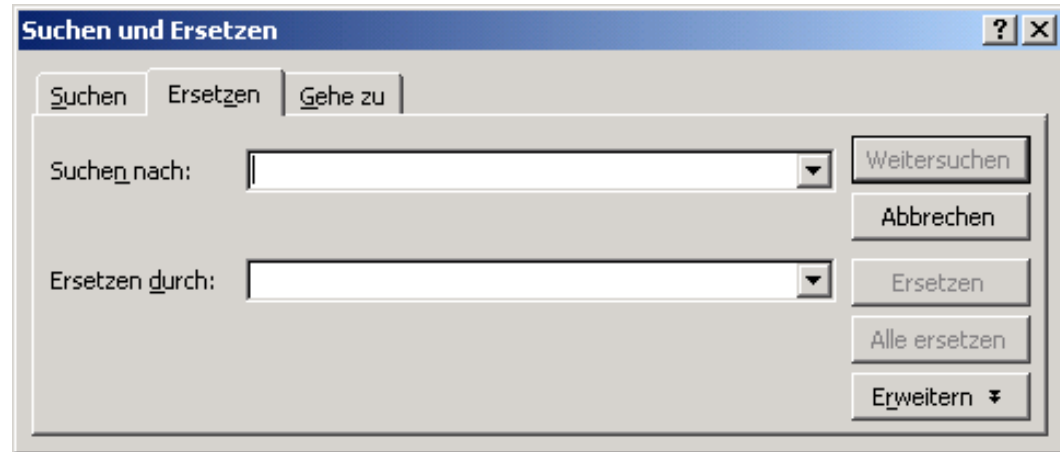
# Standarddialoge: Beispiele

Choose Color

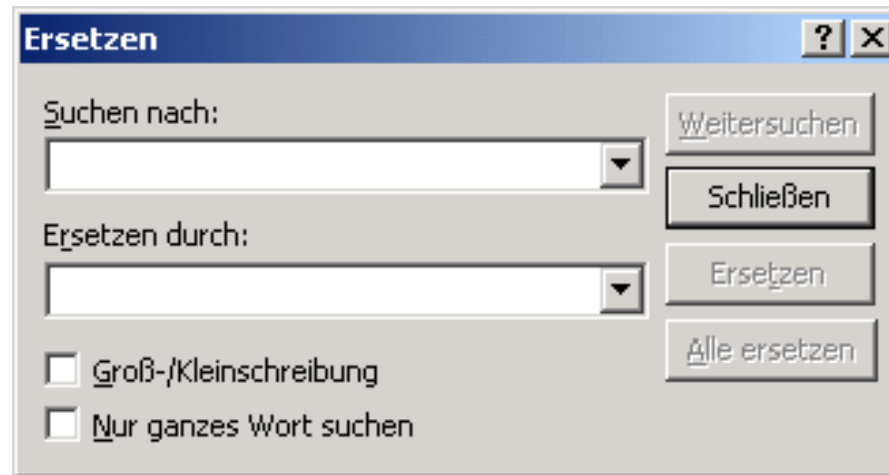


# Standarddialoge: Suchen & Ersetzen

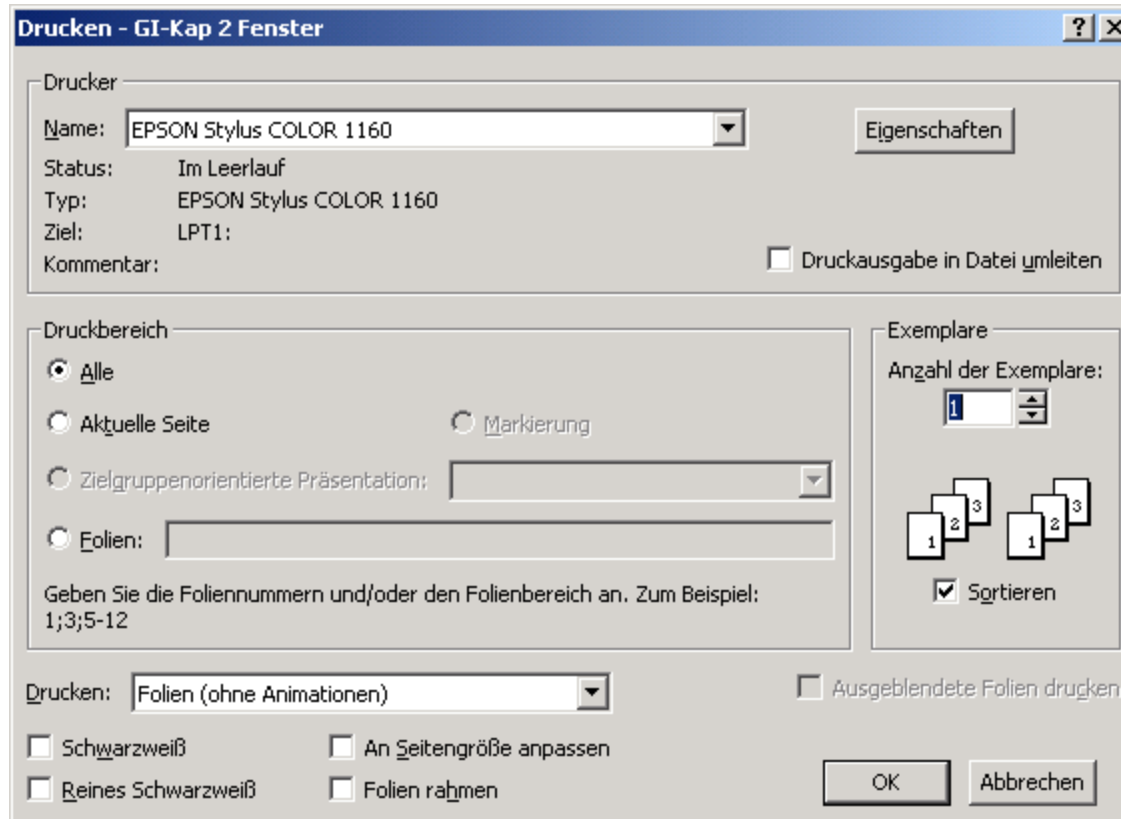
Winword:



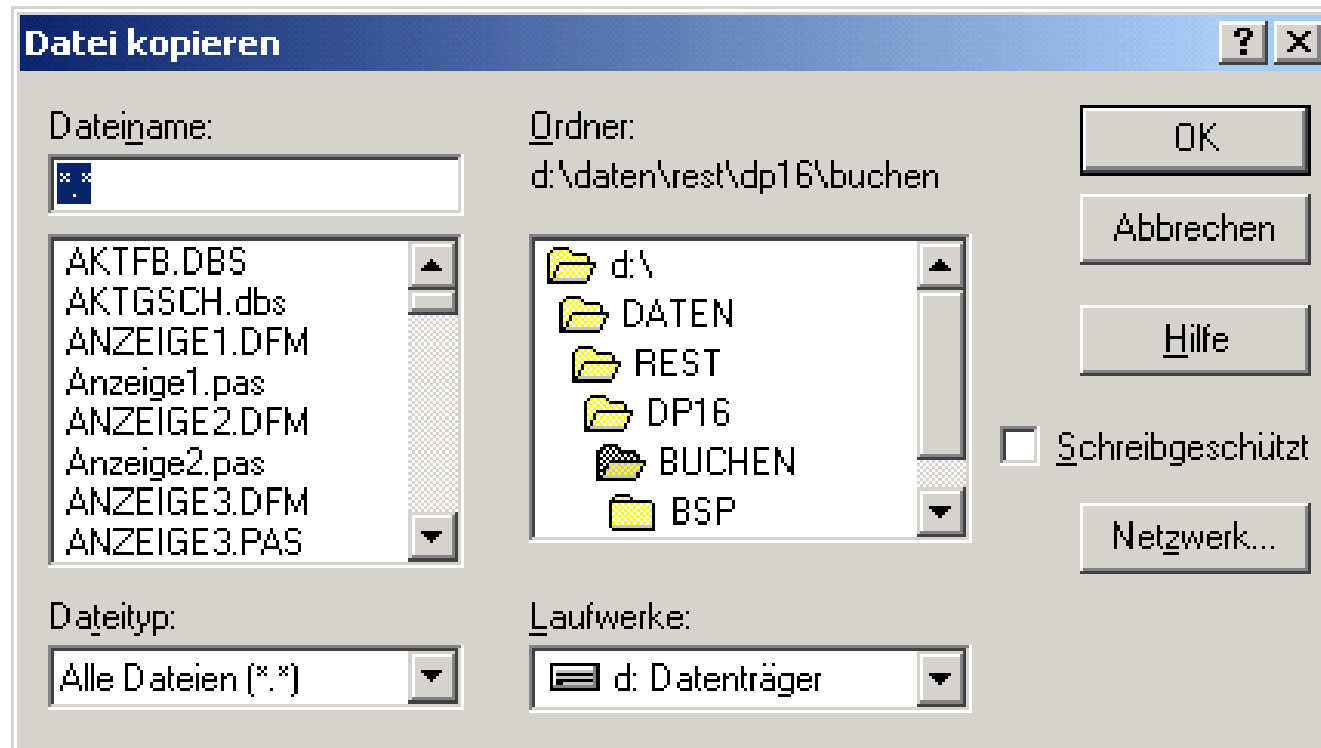
Notepad



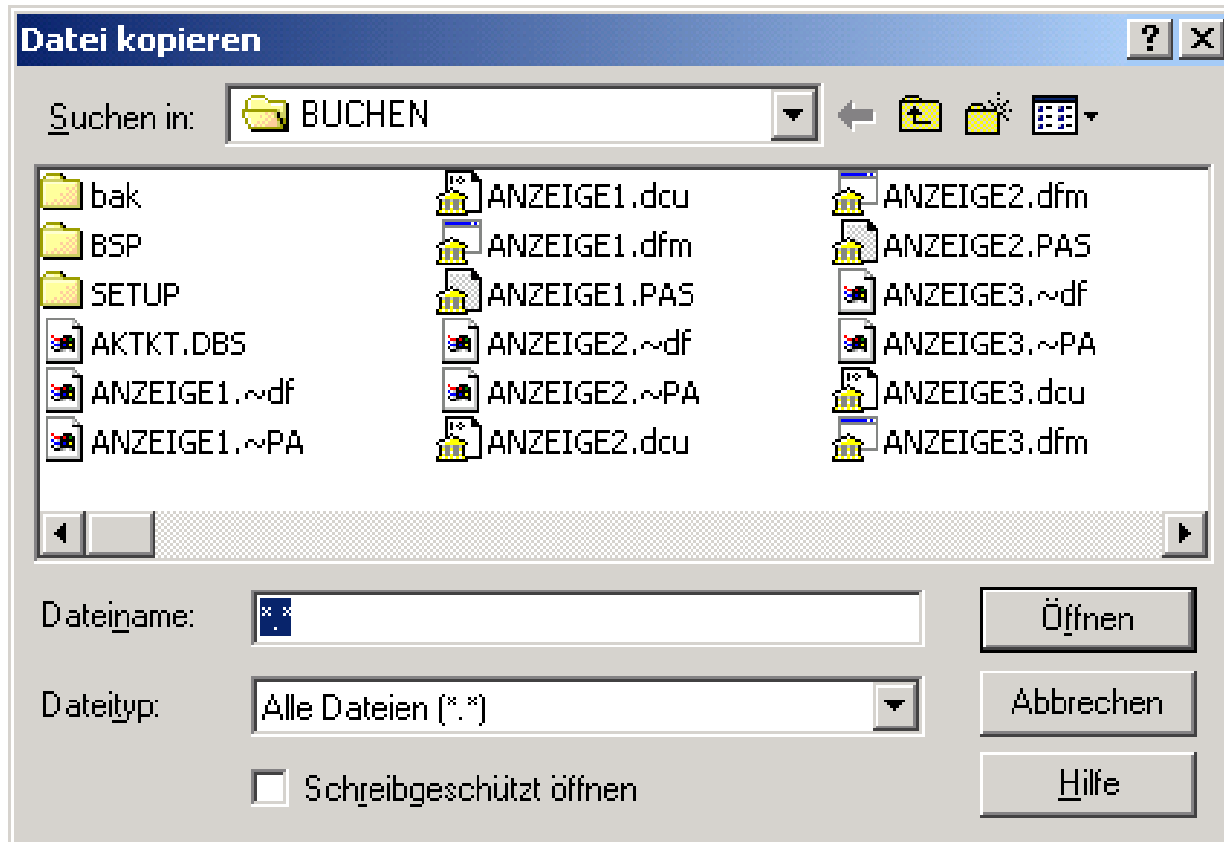
# Standarddialoge: Drucken (Powerpoint)



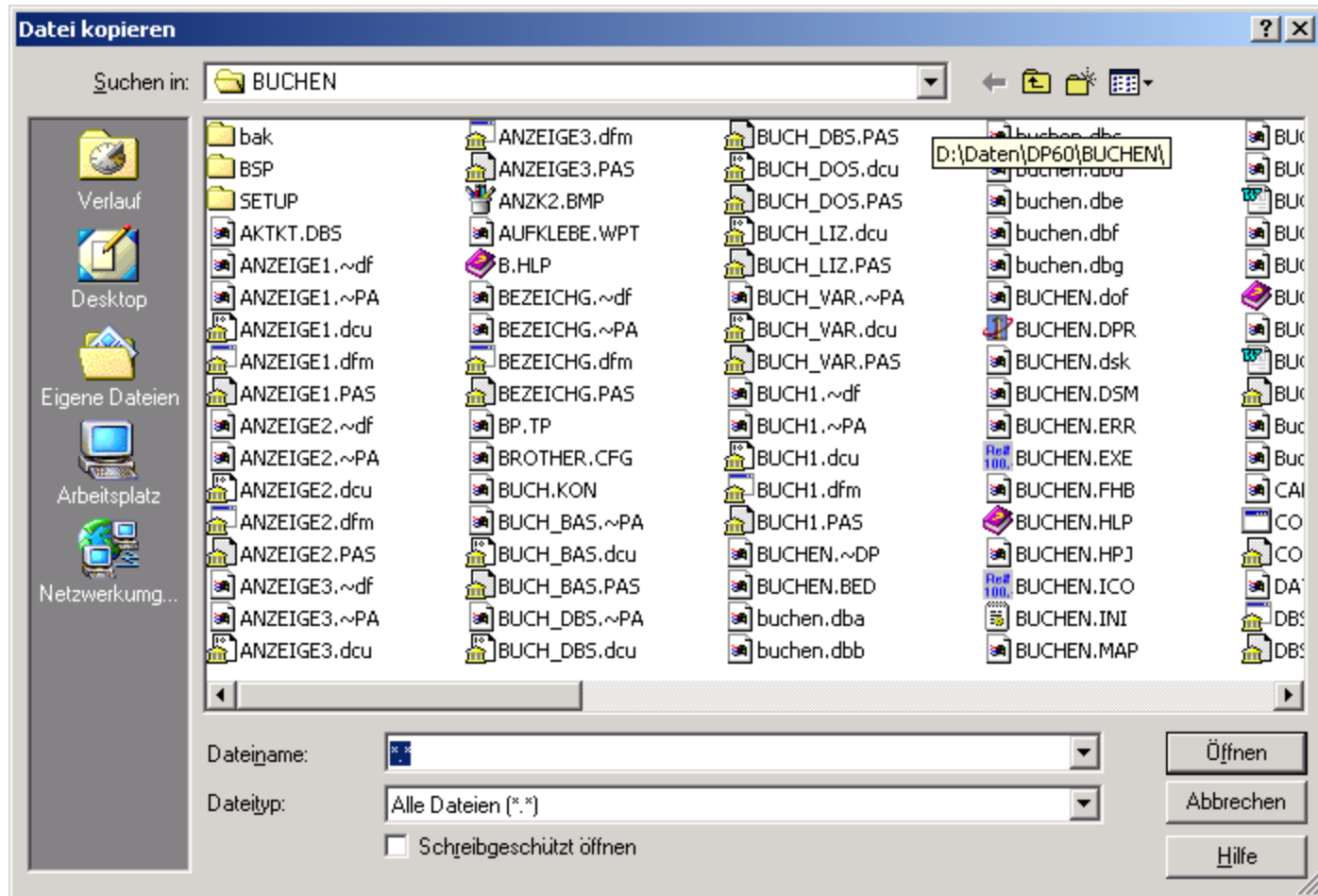
# Standarddialog Opendatei: Windows 3.1



# Standarddialog: Windows 95



# Standarddialog: Windows 2000, XP, Vista, Win 7





# Standarddialog: Aufruf (Delphi)

```
OpenDialog1.FileName:= '*.doc';
OpenDialog1.InitialDir := 'c:\eigene Dateien';
OpenDialog1.Filter:= 'DOC-Dateien (*.DOC)|*.DOC;*.DOCX|' +
                    'RTF-Dateien (*.RTF)|*.RTF|' +
                    'DOT-Dateien (*.DOT)|*.DOT|' +
                    'Alle Dateien (*.*)|*.*';

OpenDialog1.Title := 'Datei Öffnen';
OpenDialog1.Options :=
    [ofEnableSizing, ofHideReadOnly, ofAllowMultiSelect, ofPathMustExist, ofShowHelp];

if (OpenDialog1.execute) {
    writeln(OpenDialog.FileName);
}
```

# Standarddialog: Aufruf (Visual Studio)

```
CString sDefExt, sFileName, sFilter;
sDefExt = ".html";
sFilter =
    "HTML-Dateien (*.html;*.htm)|*.html;*.htm|Alle Dateien (*.*)|*.*||";
sFileName="test.html";
CFileDialog dlg(FALSE, sDefExt, sFileName,
    OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT, sFilter);
dlg.m_ofn.lpstrTitle = "Datei öffnen"; // Überschrift
if (dlg.DoModal() != IDOK)    return;

CString sOutput = dlg.GetPathName();
// Ergebnis:
AfxMessageBox( dlg.m_ofn.lpstrFile);
AfxMessageBox( sOutput);
```

# Standarddialog: Aufruf (Java)

```
void MnOpen_actionPerformed(ActionEvent e) {
    JFileChooser jFileChooser1 = new JFileChooser();
    String sFileName;
    MyFileFilter javaFilter, txtFilter;
    javaFilter = new MyFileFilter("java", "Java-Texte");
    txtFilter = new MyFileFilter( new String[] { "dat", "txt" }, "Texte");
    jFileChooser1.setMultiSelectionEnabled(false);
    jFileChooser1.addChoosableFileFilter(javaFilter);
    jFileChooser1.addChoosableFileFilter(txtFilter);
    jFileChooser1.setCurrentDirectory( new File("C:\\Daten") );
    if (JFileChooser.APPROVE_OPTION == jFileChooser1.showOpenDialog(this)) {
        // Rufen Sie openFile auf, um Text aus einer Datei in die TextArea zu laden
        sFileName = jFileChooser1.getSelectedFile().getPath();
        InternalFrame1 frame = new InternalFrame1( this, sFileName );
        insertFrame(frame,1);
    }
} // MnOpen
```

# Standarddialog: Aufruf (Java)

```
void MnOpen_actionPerformed(ActionEvent e) {
    JFileChooser jFileChooser1 = new JFileChooser();
    String sFileName;
    MyFileFilter javaFilter, txtFilter;
    jFileChooser1.setMultiSelectionEnabled(true);
    jFileChooser1.setCurrentDirectory( new File("C:\\\\Daten") );
    if (JFileChooser.APPROVE_OPTION == jFileChooser1.showOpenDialog\(this\)) {
        // Rufen Sie openFile auf, um Text aus einer Datei in die TextArea zu laden
        File[] files = jFileChooser1.getSelectedFiles();
        for (File file : files ) {
            sFileName = file.getPath\(\);
            InternalFrame1 frame = new InternalFrame1( this, sFileName );
            insertFrame(frame,1);
        }
    }
} // MnOpen
```

# Standarddialog: Aufruf (C#)

```
OpenFileDialog openFileDialog1 = new OpenFileDialog();
openFileDialog1.Filter =
    "Textdateien (*.txt;*dat)|*.txt;*.dat|pdf files (*.pdf)|*.pdf|All files (*.*)|*.*";
openFileDialog1.FilterIndex = 2;// zählt von 1 !
openFileDialog1.DefaultExt = ".pdf";
openFileDialog1.InitialDirectory="c:\\daten";
openFileDialog1.Multiselect=false;
openFileDialog1.RestoreDirectory = true;
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
    string sFile = openFileDialog1.FileName.ToString()
    MessageBox.Show(sFile, Application.ProductName);
}
```