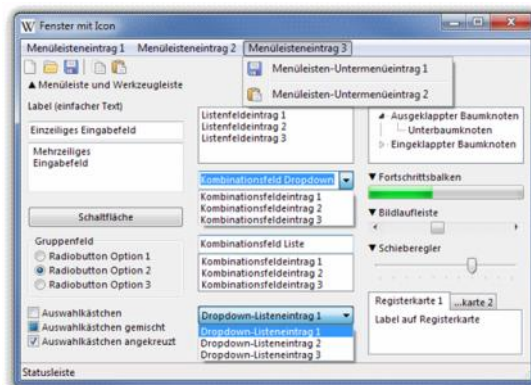


## Modul: Programmierung B-PRG Grundlagen der Programmierung 1

### V15 Entwicklung von GUIs (2)

Prof. Dr. Detlef Krömker  
Professur für Graphische Datenverarbeitung  
Institut für Informatik  
Fachbereich Informatik und Mathematik (12)

So sieht das dann (unter Windows 7) aus und wie man das macht, schauen wir uns jetzt an:



Aus:  
[de.wikipedia.org/wiki/Steuerelement](http://de.wikipedia.org/wiki/Steuerelement)

unter anderen Betriebssystemen sieht es etwas anders aus.

## Rückblick



- tkinter ... was ist das?
- Nur die allerwichtigsten Widgets und Methoden:
  - Das root-Window
  - Das Label-Widget
  - Die Layout- (Geometrie-) Manager: pack und grid
  - Das Message-Widget
  - Das Button-Widget
  - Das Entry-Widget
- Zwischen-Zusammenfassung und Ausblick

## ... und das steht heute an:

- Das Menue Widget
- Das Frame Widget
- Standardattribute für TK Widgets
- Icons statt Text
- Standarddialoge
- Das ttk-(Unter-)Modul
- Zusammenfassung

## Menues ... "nur" ein Beispielprogramm (1)

```
import tkinter as tk

class MenuDemo:
    def __init__(self):
        window = tk.Tk()
        window.title("Menu Demo")

        # Create a menu bar
        menubar = tk.Menu(window)
        window.config(menu = menubar) # Display the menu bar.
```

## Beispielprogramm (2)

```
# Create a pull-down menu, and add it to the menu bar
operation_menu = tk.Menu(menubar, tearoff = 0)
menubar.add_cascade(label = "Operation", menu = operation_menu)
# Add Menu items
operation_menu.add_command(label = "Addiere",
    command = self.add)
operation_menu.add_command(label = "Subtrahiere",
    command = self.subtract)
operation_menu.add_separator()
operation_menu.add_command(label = "Multipliziere",
    command = self.multiply)
operation_menu.add_command(label = "Dividiere",
    command = self.divide)

# Create more pull-down menus
# omitted
tk.mainloop()
```

## Beispielprogramm (3)

```
# Die Methoden

def add(self):
    print('add called')

def subtract(self):
    print('subtract called')

def multiply(self):
    print('multiply called')

def divide(self):
    print('divide called')

Demo = MenuDemo()
```

## Standardattribute für TK Widgets

- Jedes Widget hat einen Satz von Attributen, die man verändern kann.
- Hierzu gehören die Optionen, die Aussehen und Verhalten steuern, z.B.
  - Fonts
  - Farben
  - Größen
  - Beschriftungen
- Diese können durch Keyword Argumente beim Aufruf des Konstruktors gesetzt werden, z.B.
  - text = 'PANIC!'
  - height = 20
- Später sind diese jederzeit änderbar durch die `.config()`-Methode. Abfragen kann man die aktuellen Werte durch `.cget()`.

## Colors (1)

- Es gibt in Tk zwei Wege um Farben zu spezifizieren:

- Als hexadezimal String mit den Rot, Grün, Blau Anteilen der additiven Farbmischung, r,g,b ist jeweils ein Hex-Zeichen:

# rgb	4 Bit pro Kanal = 1 Hex-Wert
# rrggbb	8 Bit pro Kanal
# rrrgggbbb	12 Bit pro Kanal

z.B.    '#fff'            ist weiß,  
           '#000000'       ist schwarz,  
           '#000ffff000'   ist reines (knalliges)grün und  
           '#00ffff'        ist reines cyan

## Colors (2)

- Über Farbnamen:

z.B. 'white', 'black', 'red', 'green', 'blue', 'cyan',  
       'yellow' und 'magenta' sind überall verfügbar.

weitere Namen sind System-abhängig. VORSICHT!

## Der *relief style* eines Widgets

- ▶ Zeigt 3D-Effekte für ein Widget. So können z.B. Buttons aussehen:



- ▶ Die Weite des borders können durch die Option `borderwidth`.
- ▶ Es gibt diverse weitere Standardattribute, siehe z.B. <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/std-attrs.html> und Folgeseiten.

## Icons statt Text (1)

Bei vielen Widgets können anstelle von Text auch Bilder (meist) icons eingesetzt werden. Beispiel:

```
import tkinter as tk
root = tk.Tk()
logo = tk.PhotoImage(file = "../images/<name>.gif")
window = tk.Label(root, image=logo).pack
```

Alternative: Anstatt `PhotoImage` kann man für ein Binärbild (z.B. Schwarz/Weiß) auch nutzen:

```
tk.BitmapImage(file= "<filename> [ , background=b]
[ , foreground=c])
```

**In <filename> muss ein pbm image file enthalten sein. b und c sind Colors.**

## Icons statt Text (2)

Unterstützt werden bei `PhotoImage` folgende Bild-Fileformate:

- Graphics Interchange Format **GIF**  
8 Bit pro Pixel, indexed, erlaubt **transparenten Hintergrund**,  
ideal für Logos
- Portable Graymap (**PGM**) und
- Portable Pixmap (**PPM**) (24 Bit Pixel)

werden von anderen Werkzeugen relativ selten unterstützt!  
Konvertierung ist aber nötig!

## Standarddialoge

**Tkinter** stellt fertig ausgestaltete Standarddialoge für häufig vorkommende Situationen zur Verfügung.

Diese sind:

- **Meldungen** (*Message Boxes*)
- Dateiauswahl (*Open File Dialogue*)
- Farbauswahl (*Color Chooser*)

### Modul

`tkinter.messagebox`  
`tkinter.filedialog`  
`tkinter.colorchooser`

Diese Dialoge werden in zugehörigen Modulen bereitgestellt,  
die ggf. zusätzlich zu Tkinter zu importieren sind.



**Weitere Standarddialoge** sind: `tkinter.commondialog`, `tkinter.simpledialog`,  
`tkinter.font`, `tkinter.dnd`, `tkinter.scrolledtext`, `tkinter.tix`, `tkinter.ttk`

## Meldungen (Message Boxes) (1)

```
import tkinter as tk
import tkinter.messagebox as mb

mb.askokcancel('okcancel', 'Sollen wir wirklich ...')
mb.askquestion('question', 'Hier ist die Frage\nWollen ...')

if mb.askretrycancel('retrycancel', 'Sollen wir noch einmal?',
                    default=mb.RETRY):
    print('Toll')
# the following default-constants are defined in tkinter.messagebox
# CANCEL, IGNORE, OK, NO, RETRY, YES
```

Es werden insgesamt 7  
verschiedene  
Meldungen unterstützt:



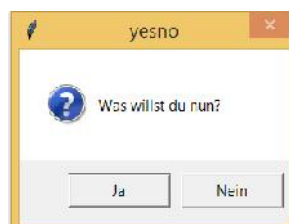
15

Programmieren 1 / EPR

Prof. Dr. Detlef Krönker

## Meldungen (Message Boxes) (2)

- if mb.askyesno('yesno', 'Was willst du nun?'):
  - print('o.k.!')
  - else:
  - print('Auch o.k.!')
- # All Methods above return True or False.



16

Programmieren 1 / EPR

Prof. Dr. Detlef Krönker



## Meldungen (Message Boxes) (3)

# Die following three Methods return 'ok' after pressing the ok-button.

```
mb.showwarning('showwarning', 'Aufpassen, biitel')
mb.showerror('showerror', 'Hat leider nicht geklappt!')
mb.showinfo('showinfo', 'Ist halt so, wie es ist.', icon = mb.WARNING)
```

# the following icon-constants are defined in tkinter.messagebox  
# ERROR, INFO, QUESTION, WARNING



17

Programmieren 1 / EPR

Prof. Dr. Detlef Krönker

## Das ttk (Unter-)Modul

- Ab Tk 8.5 steht allgemein das ttk (Unter-)Modul zur Verfügung.
- Dieses Modul überschreibt viele Original TK-Widgets.
- Hauptziel war:
  - Plattform-spezifische Erscheinung (die Original TK-Widgets weichen davon ab.)
  - ttk erlaubt es die Anwendungen in einem generisches Weg zu schreiben und sie erscheinen dann doch wie eine Windows-, MacOS-Applikation (ohne Änderung der Quelle).
  - Jede mögliche Erscheinungsform wird durch das benannte ttk theme identifiziert.

18

Programmieren 1 / EPR

Prof. Dr. Detlef Krönker

## Importieren von ttk

- ▶ Bitte **immer** so importieren:

```
import tkinter as tk
from tkinter import ttk
```

Dann:

```
b1 = tk.Button(...) # uses a standard tk button
b2 = ttk.Button(...) # uses a ttk button
```

~~Explizit:~~

~~from Tkinter import \*~~  
~~from ttk import \*~~

ist durch unsere Konventionen  
verboten!

## Zusammenfassung (1)

- ▶ Ich hoffe, auch Sie haben insgesamt Spaß gehabt und ein gutes Gefühl, obwohl wir längst nicht alles behandeln konnten.
- ▶ Leider müssen Sie einiges noch nacharbeiten, da uns in der Vorlesung die Zeit fehlte.
- ▶ Es bieten sich an: Das "TKinter-Tutorial" von Rolf Klein:  
[http://www.python-kurs.eu/python\\_tkinter.php](http://www.python-kurs.eu/python_tkinter.php) (deutsch)  
dies war zum Teil auch die Vorlage für diese Vorlesung (ist aber leider nicht vollständig!)
- ▶ Die "Tkinter 8.5 reference: a GUI for Python" von John W. Shipman.  
<http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>  
(sieht sehr vollständig aus).

## Fragen und (hoffentlich) Antworten Zusammenfassung (2)

- Es gibt tatsächlich noch sehr viel mehr Widgets und Standarddialoge
- Auch einzelne Events (keystrokes, Mouse-Buttons, ...) können als Events abgefangen werden.
- Trotz der relativ intensiven Unterstützung durch Tkinter sind die Programme immer länger als CLI-Interfaces!
- Leider: Viel Arbeit ... was hatte ich gesagt: 50% der Gesamtkosten.
- Es gibt zu GUIs auch ein Quiz!

## Ausblick

... weiter geht es am **Montag, den 11.12. 2017.**

Thema:

**V16 OO – Analyse und Design**

**Und ... weiterhin viel Erfolg**