



Informatik Klasse 13, Foliensatz 3

Hallo Tkinter

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal
24. August 2009

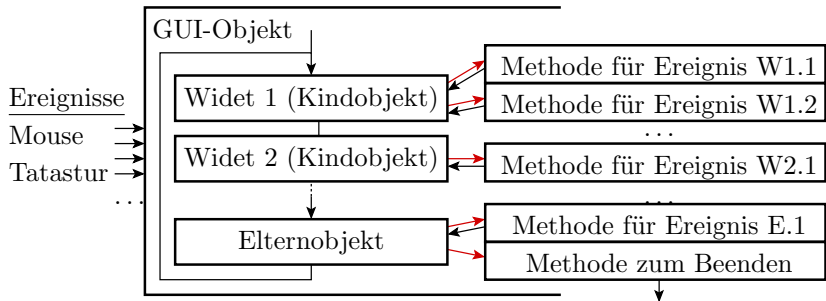


Experiment: Tutorial Chapter 2

```
1: # File: hello1.py
2: from Tkinter import *
3: root = Tk()
4: w = Label(root, text="Hello, world!")
5: w.pack()
6: root.mainloop()
```

- 1 Kommentar
- 2 Import der Klassen TK und Label
- 3 Erzeugung eines GUI- (graphical user interface) Objekts (root widget)
- 4 Erzeugen eines Textfeldes (label widget)
- 5 Aufruf der Anordnungsmethode des Labels
- 6 Start der Ereignisschleife des GUI-Objekts

Widget: Elemente des graphischen Nutzerinterfaces



→ Aufruf, wenn das zugeordnete Ereignis eingetreten ist

- Es gibt kein Hauptprogramm, sondern nur Methoden, die bei bestimmten Ereignissen gestartet werden

GUI-Programmierung:

- Zusammensetzen einer Oberfläche aus Widgets
- Konfiguration über Attribute
- Programmieren der anwendungsspezifischen Methoden



Experiment: Tutorial, Chapter 3

```
from Tkinter import *
class App:
    def __init__(self, master):
        frame = Frame(master)
        frame.pack()
        self.button = Button(frame, text="QUIT", fg="red",
                              command=frame.quit)
        self.button.pack(side=LEFT)
        self.hi_there = Button(frame, text="Hello",
                                command=self.say_hi)
        self.hi_there.pack(side=LEFT)
    def say_hi(self):
        print "hi there, everyone!"

root = Tk()
app = App(root)
root.mainloop()
```



- Eine Applikationsklasse besteht aus
 - einem Konstruktor zur Erzeugung der Widgets
 - eigenen Methoden, die Ereignisse zugeordnet sind
- Oberflächen sind hierarchische Objekte mit einer »enthält Teilfenster«-Relation:
 - Applikation »self« enthält den Rahmen »frame«
 - Der Rahmen »frame« enthält die Schaltflächen »self.button« und »self.hi_there«
 - Das Elternobjekt ist das erste Argument im Konstruktor
- alle anderen Argumente im Konstruktor sind Attribute des Widget, den Werte zugeordnet werden, z.B. die Farbe
- »command=Methode« bindet eine Aktion an das Ereignis »Tastenbetätigung«
- »pack()« Anordnungsmethode; für alle Widgets erforderlich



Aufgabe 3.1: Hallo Welt

Erweitern Sie das Programm »Hallo Welt« um einen »Button-Widget«, mit dem das Programm beendet werden kann.



Tastensfeld zum Start von Programmen

Schreiben Sie eine Applikationsklasse mit einem Konstruktor und vier anwendungsspezifischen Methoden.

- Der Konstruktor soll in einem Rahmen vier Tasten und ein Label erzeugen und untereinander anordnen.
- Die insgesamt fünf Widgets sind mit »Taste 1«, »Taste 2«, »Taste 3«, »Taste 4« und »Textfeld« zu beschriften.
- Die bei Tastenbetätigung auszuführenden Methoden sollen auf der Standardausgabe den Text »Taste i gedrückt« ausgeben (i – Tastennummer)



Oberfläche mit Textausgabe

Mit der Methode »config« können für eine Widget nachträglich die Attribute verändert werden, z.B. für

```
self.Ausgabertext = Label(parent, text='Originaltext')
```

der dargestellte Text

```
self.Ausgabertext.config(text='neuer Text')
```

Schreiben Sie das vorherige Programm so um, dass die Ausgabe zusätzlich im »Label-Widjet« erfolgt.