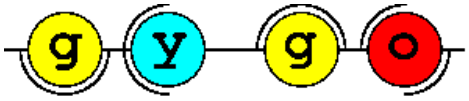
Gui-Objekte und Datenmodellobjekte

Anja Faatz

2010



Gymnasium Gonsenheim

Die Themen

- Beispiel Ampel
- Trennung: Datenmodell-GUI
- Datenmodell
- GUI
- Erzeugung der Objekte
- Übungen
- Literatur

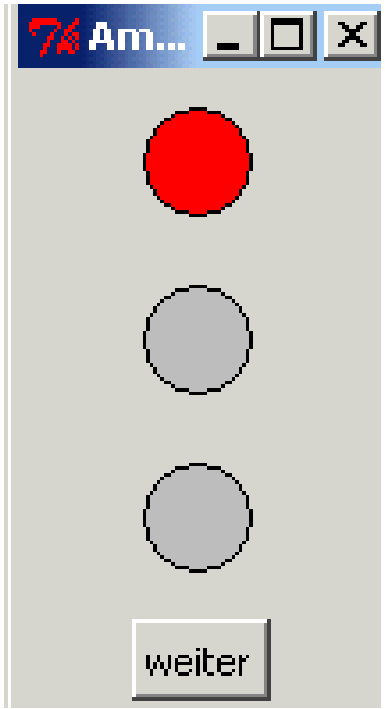
Die Themen

- Beispiel Ampel
- Trennung: Datenmodell-GUI
- Datenmodell
- GUI
- Erzeugung der Objekte
- Übungen
- Literatur

Beispiel Ampel

Es soll eine GUI für eine Ampel geschrieben werden.

Die Ampel wird über einen Button weitergeschaltet.



Beispiel Ampel

```
...
ampelFenster = Tk()
ampelFenster.title('Ampel')

zeichenflaeche=Canvas(ampelFenster,width=50,height=150)
buttons=Button(ampelFenster,text="weiter",command=buttonWeiterschalten)
zeichenflaeche.pack()
buttons.pack()

rot = zeichenflaeche.create_oval(10,10,40,40,fill='red')
gelb = zeichenflaeche.create_oval(10,60,40,90,fill='grey')
gruen = zeichenflaeche.create_oval(10,110,40,140,fill='grey')

ampelFenster.mainloop()
```

Fenster erzeugen

Zeichenfläche und Button erzeugen

drei Lampen zeichnen

starten

Beispiel Ampel

```
from tkinter import *

def buttonWeiterschalten():
    colour1=zeichenflaeche.itemcget(rot, 'fill')
    colour2=zeichenflaeche.itemcget(gelb, 'fill')
    colour3=zeichenflaeche.itemcget(gruen, 'fill')
    if colour1=='red' and colour2=='grey':
        Zaehler=0
    elif colour1=='red' and colour2=='yellow':
        Zaehler=1
    elif colour3=='green':
        Zaehler=2
    else: Zaehler=3
    Zaehler = (Zaehler + 1) % 4
    Farben = ['Rot','Rot/Gelb','Gruen','Gelb']
    Zustand = Farben[Zaehler]
...
```

Verwaltung der Daten

Verarbeitung der Daten

0 → 1 → 2 → 3 → 0 → ...

Zaehler auswerten

Beispiel Ampel

```
...

    if 'Rot' in Zustand:
        zeichenflaeche.itemconfig(rot,fill='red')
    else:
        zeichenflaeche.itemconfig(rot,fill='grey')
    if 'Gelb' in Zustand:
        zeichenflaeche.itemconfig(gelb,fill='yellow')
    else:
        zeichenflaeche.itemconfig(gelb,fill='grey')
    if 'Gruen' in Zustand:
        zeichenflaeche.itemconfig(gruen,fill='green')
    else:
        zeichenflaeche.itemconfig(gruen,fill='grey')
...
```

Ampel zeichnen

Beispiel Ampel

Nachteile

- Programm läuft nicht ohne GUI
- GUI-Objekte sind nicht für die Datenhaltung einer Anwendung gedacht

```
colour1=zeichenflaeche.itemcget(rot, 'fill')
```

aktueller Zustand der roten Lampe

- Programmerweiterung unübersichtlich

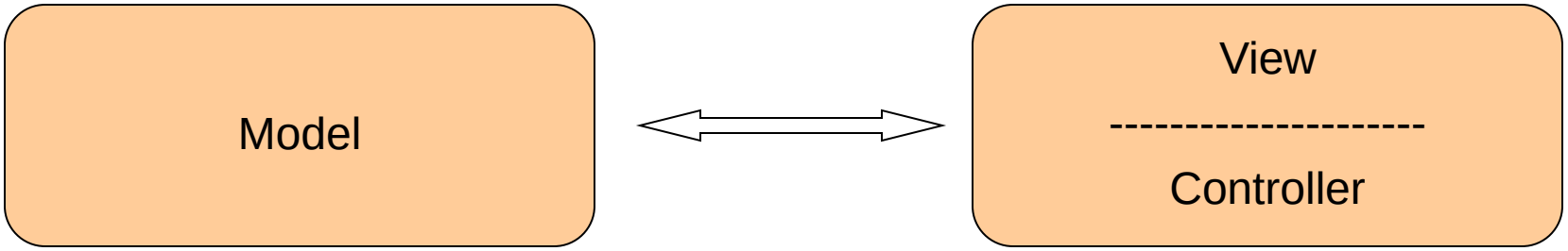
Die Themen

- Beispiel Ampel
- Trennung: Datenmodell-GUI
- Datenmodell
- GUI
- Erzeugung der Objekte
- Übungen
- Literatur

Trennung: Datenmodell-GUI

„Ein grundlegendes Prinzip beim Entwurf von Software-Systemen besteht heute in der klaren Trennung zwischen Benutzeroberfläche und Fachkonzept“

(H.Balzert: Lehrbuch der Informatik, S. 123)



http://www.oszhandel.de/gymnasium/faecher/informatik/didaktik/delphi-java.pdf

Trennung: Datenmodell-GUI

Model = Applikation ohne Benutzeroberfläche
interne Datenverarbeitung

View = verantwortlich für die aktuelle Darstellung der Eingangs-/Ausgangsdaten der Anzeigenobjekte

Controller = überwacht und kontrolliert alle Eingabegeräte

View und Controller bilden zusammen die Benutzer-Oberfläche.

http://www.oszhandel.de/gymnasium/faecher/informatik/didaktik/delphi-java.pdf

Trennung: Datenmodell-GUI

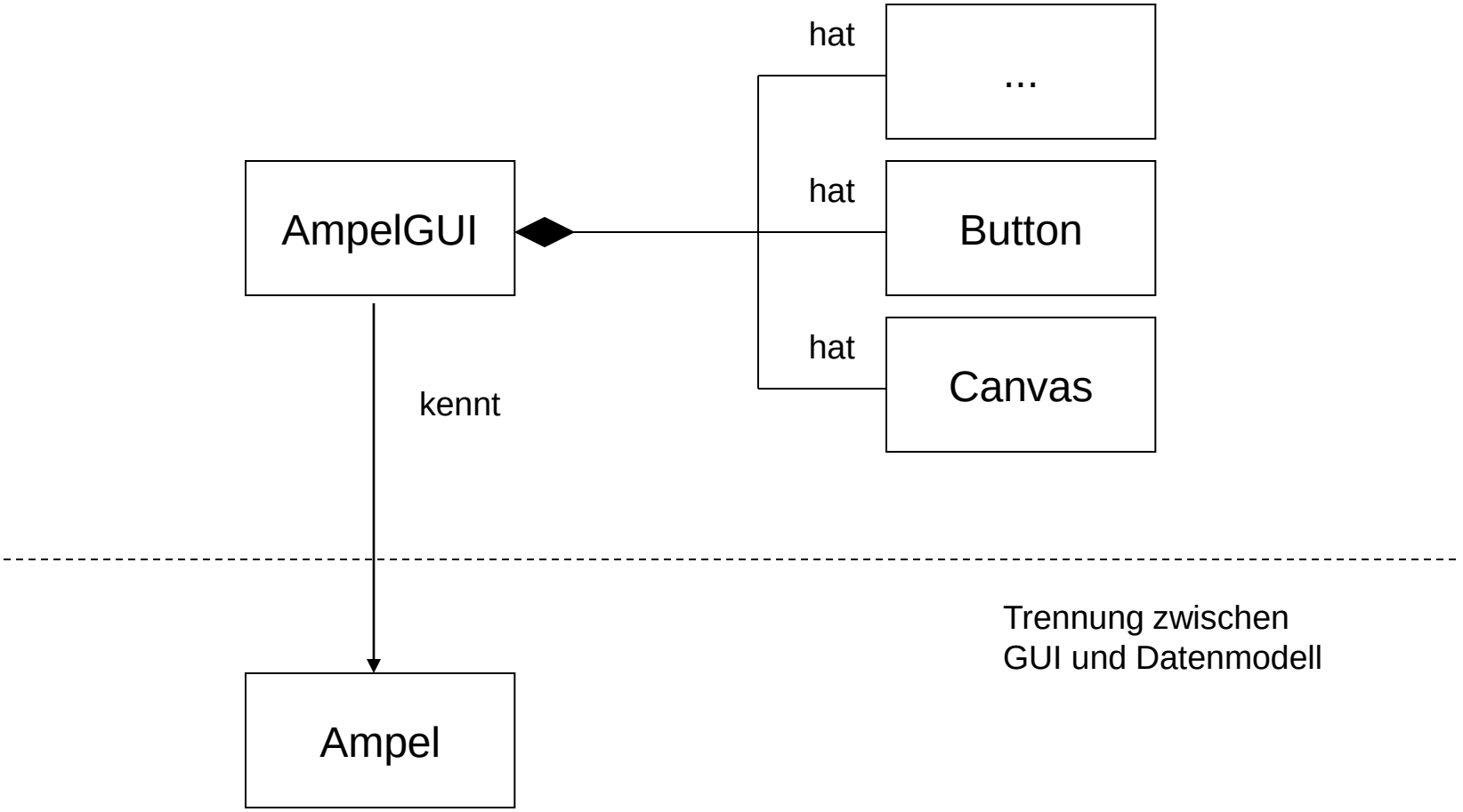
- Zu dem reinen Datenobjekt kommen GUI-Objekte (Fenster, Label, Button,...) hinzu
- **Model** weiß nichts über **View-Contoller**
- **View-Controller** kennen **Model**, holen von und schicken ihm Daten

Die interne Datenverarbeitung ist gänzlich von der Benutzeroberfläche abgekoppelt.

Änderungen der Benutzeroberfläche haben keinen Einfluss auf die Datenverarbeitung und Datenstruktur.

http://www.oszhandel.de/gymnasium/faecher/informatik/didaktik/delphi-java.pdf

Trennung: Datenmodell-GUI



Die Themen

- Beispiel Ampel
- Trennung: Datenmodell-GUI
- Datenmodell
- GUI
- Erzeugung der Objekte
- Übungen
- Literatur

Datenmodell

```
class Ampel(object):

    def __init__(self):
        self.zaehler = 0
        self.zustand='Rot'

    def weiterschalten(self):
        self.zaehler = (self.zaehler + 1) % 4
        Farben = ['Rot','Rot/Gelb','Gruen','Gelb']
        self.zustand = Farben[self.zaehler]
```

Konstruktor

Attribute zur Verwaltung der Ampeldaten mit Anfangswerten

Ampel weiterschalten

Erzeugt neue Werte für die Attribute zur Verwaltung der Ampel

Datenmodell

```
>>> a=Ampel()
>>> a.zustand
'Rot'
>>> a.weiterschalten()
>>> a.zustand
'Rot/Gelb'
```

Wenn man diese Klassendeklaration ausführt, dann lässt sich folgender Python-Dialog zur Simulation einer Ampel führen

Wichtig: Die Klasse Ampel ist so konzipiert, dass sie ohne grafische Oberfläche benutzt werden kann.

Sie liefert also ein reines Datenmodell.

Die Themen

- Beispiel Ampel
- Trennung: Datenmodell-GUI
- Datenmodell
- GUI
- Erzeugung der Objekte
- Übungen
- Literatur

GUI

```
class AmpelGUI(object):

    def __init__(self, a):
        self.ampel = a
        self.ampelFenster = Tk()
        self.ampelFenster.title('Ampel')
        self.zeichenflaeche=Canvas(self.ampelFenster,width=50,height=150)
        self.buttons=Button(self.ampelFenster,text="weiter",command=self.buttonWeiterschalten)
        self.zeichenflaeche.pack()
        self.buttons.pack()
        self.rot = self.zeichenflaeche.create_oval(10,10,40,40,fill='red')
        self.gelb = self.zeichenflaeche.create_oval(10,60,40,90,fill='grey')
        self.gruen = self.zeichenflaeche.create_oval(10,110,40,140,fill='grey')
    ...
```

Konstruktor

Fenster erzeugen

Zeichenfläche und Button erzeugen

drei Lampen zeichnen

GUI

```
...
    def ampelZeichnen(self):
        if 'Rot' in self.ampel.zustand:
            self.zeichenflaeche.itemconfig(self.rot,fill='red')
        else:
            self.zeichenflaeche.itemconfig(self.rot,fill='grey')
        if 'Gelb' in self.ampel.zustand:
            self.zeichenflaeche.itemconfig(self.gelb,fill='yellow')
        else:
            self.zeichenflaeche.itemconfig(self.gelb,fill='grey')
        if 'Gruen' in self.ampel.zustand:
            self.zeichenflaeche.itemconfig(self.gruen,fill='green')
        else:
            self.zeichenflaeche.itemconfig(self.gruen,fill='grey')
...
```

Anzeige der Daten

GUI

```
...
    def buttonWeiterschalten(self):
        self.ampel.weiterschalten()
        self.ampelZeichnen()
```

Ereignisverarbeitung

Mit Hilfe des Datenmodell-Objektes ampel wird gemeinsam mit den GUI-Objekten die Aufgaben der Benutzeroberfläche erledigt.

Die Datenhaltung und Datenverarbeitung wird komplett von der Klasse Ampel übernommen.

Die Themen

- Beispiel Ampel
- Trennung: Datenmodell-GUI
- Datenmodell
- GUI
- Erzeugung der Objekte
- Übungen
- Literatur

Erzeugung der Objekte

```
from Ampeldatenmodell1 import *
a = Ampel()

from AmpelGUI import *
gui = AmpelGUI(a)
gui.ampelFenster.mainloop()
```

Datenmodell

Ampel-Objekt kennt AmpelGUI-Objekt nicht

GUI-Objekte

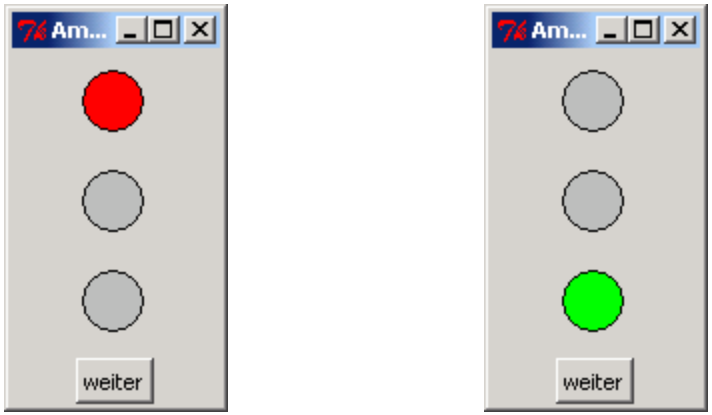
AmpelGUI-Objekt kennt Ampel-Objekt

Die Themen

- Beispiel Ampel
- Trennung: Datenmodell-GUI
- Datenmodell
- GUI
- Erzeugung der Objekte
- Übungen
- Literatur

Ampel

Ändern Sie das Projekt Ampel in eine Fussgänger-Ampel.

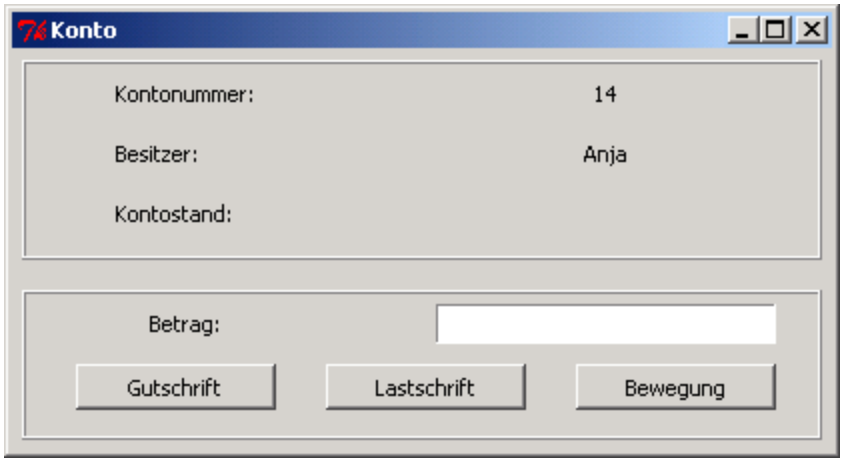


Was müssen Sie dazu alles ändern?

Konto

Schreiben Sie zu dem Datenmodell in konto.py folgende GUI.

- Button_Gutschrift: eingegebener Betrag wird zum Kontostand addiert und der Kontostand wird angezeigt
- ...



Gestalten Sie zuerst die reine Benutzungsoberfläche. Funktionen und Methoden, die an Buttons gebunden sind werden erst einmal mit leeren Anweisungen versehen. (kontoGUI1vorlage.py)

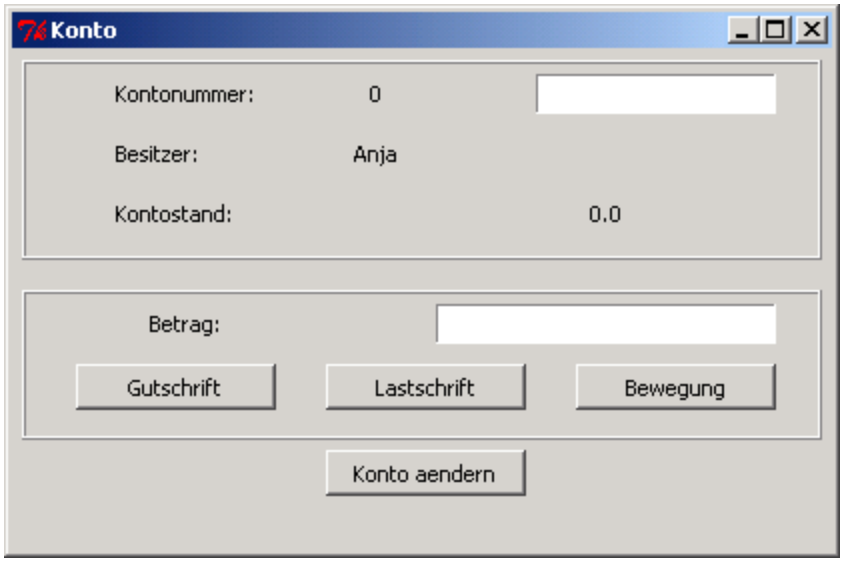
Im zweiten Schritt werden die Methodendefinitionen ausformuliert. (kontoGUI1solo.py)

Konto

Schreiben Sie zu dem Datenmodell in konto.py folgende GUI.

- Button_Gutschrift:
  eingegebener Betrag wird zum Kontostand addiert und der Kontostand wird angezeigt
- ...

- Button_Konto:
  eingegebene Kontonummer wird in einer Kontoliste gesucht und Nummer, Besitzer und aktueller Kontostand wird in den entsprechenden Labels angezeigt



Die Themen

- Beispiel Ampel
- Trennung: Datenmodell-GUI
- Datenmodell
- GUI
- Erzeugung der Objekte
- Übungen
- Literatur

Literatur

- Michael Weigend: Objektorienrierte Programmierung mit Python, 3.Auflage (mitp 2006)
- www.inf-schule.de/informatik/gui/entwicklung
- www.inf-schule.de/informatik/konzepteoop/gui/einstieg_konto
- www.oszhandel.de/gymnasium/faecher/informatik/didaktik/delphi-java.pdf