Skolan för Datavetenskap och kommunikation

Programmeringsteknik

Föreläsning 17

Idag

- Mera om GUI med Tkinter:
  - Lambda-funktioner
  - Layout - grid
  - Attributet variable
- Felhantering, exempel

Hur var det man kopplade ihop en knapp med en funktion?

```
from Tkinter import *
def byttext():
    knapp["text"] = "Aj!"
roten = Tk()
knapp = Button(roten,
               text = "Tryck inte",
               command = byttext)
knapp.pack()
roten.mainloop()
```

Parametrar då?

- Här skickar man bara med *namnet* på funktionen.
- Hur ska man göra om funktionen har parametrar?
- Lösning: Använd en lambda-funktion

Lambda-funktion…

- I Python kan man definiera en anonym funktion med följande syntax:

  lambda parametrar: uttryck

- Vanlig funktion:

```
def dubbla(x):
    return 2*x
```

- Med lambda:

```
dubbla = lambda(x) : 2*x
```

…som parameter till Button

```
from tkinter import *

def byttext(t):
    knapp["text"] = t

roten = Tk()
knapp = Button(roten,
    text = "Tryck inte",
    command = lambda: byttext("Aj!"))
knapp.pack()
roten.mainloop()
```

Layout

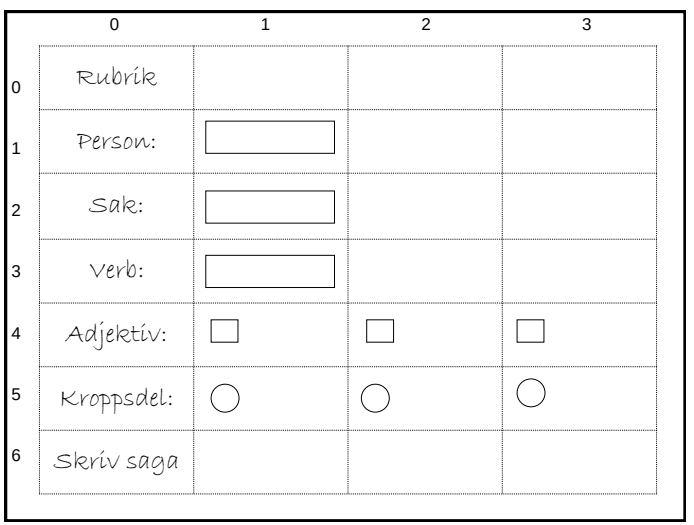
- Komponenter har metoder som styr hur de ska placeras i fönstret.
- Enklast är att använda pack:

```
knapp.pack()
```

- Men bättre kontroll fås med grid:

```
knapp.grid(row=4, column=3)
```

- Rita först en skiss över hur det ska se ut!
- Se programexemplet saga.py



.py eller .pyw?

- Spara programmet med filändelsen ".pyw", t ex "saga.pyw" så kan du starta det med dubbelklick på ikonen.

Variabler

- Ett attribut som alla komponenter har är `variable`.
- Om man i förväg skapat ett variabelobjekt:

```
s = StringVar()
```

- så kan man koppla ihop variabel och komponent med

```
knapp["variable"] = s
```

- Metoden `get` hämtar data från en variabel.

Felhantering, t ex

- Felaktig inmatning:
  - Tecken istället för tal
  - För stort/för litet tal
- Filer:
  - Infil saknas
  - Felaktiga data i filen
- Lista/dictionary:
  - Index saknas
  - Nyckel saknas

Exception - repetition

- När något blir fel i ett Python-program uppstår ett särfall, t ex NameError:

```
>>> print sko

Traceback (most recent call last):
  File "<pyshell#17>", line 1, in -toplevel-
    print sko
NameError: name 'sko' is not defined
```

- Man kan ta hand om särfall genom att införa `try-except-else`-satser för de delar i programmet som kan krascha.

Särfall - exempel

```
try:
    tal = int(input("Ge ett tal: "))
    invers = 1.0/tal
except (ZeroDivisionError):
    print("Noll kan inte inverteras")
except (ValueError):
    print("Måste vara ett tal!")
else:
    print("Inversen blev", invers)
```

Exempel i slinga

```
def lasPengar():
   """ Läser in tills man ger ett heltal"""
   pengar = None
   while not pengar:
      try:
        svar = input("Ange belopp: ")
        pengar = int(svar)
      except ValueError:
        print("Felaktigt belopp, försök igen")
   return pengar
```

Felhantering i tkinter

- I messagebox finns "popupfönster" som lämpar sig för felhantering:
  - showinfo
  - showwarning
  - showerror
  - askquestion
  - askyesnocancel
- Alla tar två parametrar: title och message
- Vissa har returvärde (askyesnocancel)

Exempel: showerror

```
from tkinter import *
rot = Tk()
messagebox.showerror(title="Fel",    \
        message="Du har just gjort fel.")

rot.mainloop()
```