

Skolan för Datavetenskap och kommunikation

Föreläsning 10

Programmeringsteknik

Vad händer nu?

Nästa vecka

- Labbtiden: uppsamling labbar/läxförhör
- förel + övn: repetition inför provet

Veckan därpå

- prov 3:e mars
- val av P-uppgifter

kortGUI0.py

```
from tkinter import *

roten = Tk()
gifbild = PhotoImage(file="ruter4.gif")
knapp = Button(roten, image=gifbild)
knapp.bild = gifbild
knapp.pack()
roten.mainloop()
```

Kortgui3.py

```
from tkinter import *

class Bildknapp(Button):

    def __init__(self, master, bildfil1, bildfil2):
        self.framsida = PhotoImage(file = bildfil1)
        self.baksida = PhotoImage(file = bildfil2)
        Button.__init__(self, master, image = self.baksida, command = self.vänd)
        self.bild = self.baksida
        self.framsidanUpp = False

    def vänd(self):
        if self.framsidanUpp:
            self.framsidanUpp = False
            self.configure(image = self.baksida)
            self.bild = self.baksida
            print("baksida")
        else:
            self.framsidanUpp = True
            self.configure(image = self.framsida)
            self.image = self.framsida
            print("framsida")

roten = Tk()
knapp = Bildknapp(roten, "ruter4.gif", "baksida.gif")
knapp.pack()
roten.mainloop()
```

Mer att lära?

Kap 1-9 i boken: grunderna i Python

Problemlösning

Arbetsätt

Prototyper

Grafiskt gränssnitt GUI med tkinter

Problemlösning

Börja med att tänka på:

- Hur ska det se ut när man kör programmet?
- Hur ska programmet arbeta? *Algoritm*
- Hur ska data lagras i programmet? *Datastruktur*
- Skriv lite i taget! *Prototyper*

Kortspelsprogram

- Blackjack
- Poker
- Finns i sjön

Algoritm

- Skapa en kortlek
- Blanda korten
- Fråga efter antal spelare
- Dela ut fem kort till varje spelare

Datastruktur

Hur ska vi representera kortleken i programmet?

Lista med Kort-objekt!

valör <input type="text" value="4"/>	valör <input type="text" value="7"/>	valör <input type="text" value="8"/>	valör <input type="text" value="9"/>
färg <input type="text" value="Hjärter"/>	färg <input type="text" value="Spader"/>	färg <input type="text" value="Hjärter"/>	färg <input type="text" value="Spader"/>
framsidan <input type="text" value="True"/>	framsidan <input type="text" value="True"/>	framsidan <input type="text" value="True"/>	framsidan <input type="text" value="True"/>

prototyper

Försök inte skriva allt på en gång!

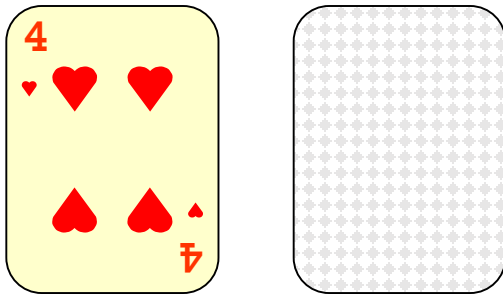
Arbeta med prototyper: körbara delversioner av programmet
(kortspel0.py, kortspel1.py, kortspel2.py, etc)

Tips: OK att skriva tomma funktioner

```
def gruyere(volym, tid):  
    pass
```

Spelkort

- Klassen Kort representerar ett spelkort



- Klassens attribut är
 - `valör` (2-10, knekt, dam, kung, ess),
 - `färg` (Klöver, Ruter, Hjärter och Spader)
 - `framsidanUpp` (True om kortet är uppvänt)
- Metoderna är
 - `__init__` (konstruktorn),
 - `__str__` (kortet som sträng),
 - `vänd` (vänder)

defaultparametrar

Vill du kunna utelämna en parameter i anropet?

Ge den då ett värde i parameterlistan.

```
def linje(längd, tecken = "-"):  
    print(tecken*längd)
```

```
linje(60)
```

```
linje(60, "*")
```

```
linje(längd = 60, tecken = "*")
```

Klass-attribut

Ett attribut som definieras i klassen, men utanför `__init__`, kallas för ett *klass-attribut*.

Varje objekt har sin egen uppsättning attribut, men det finns bara ett exemplar av klass-attributet.

```
class Tal:  
  
    klassAttribut = 17  
  
    def __init__(self):  
        self.attribut = 42
```

```
objekt = Tal()  
  
print(objekt.attribut)  
  
print(Tal.klassAttribut)
```