

Föreläsning 10

PROGRAMMERINGSTEKNIK

Kortspel – exempel med

- Flera klasser
- Arv
- Default-parametrar

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, bildfill, bildfil2):
        self.framsida = PhotoImage(file = bildfill)
        self.baksida = PhotoImage(file = bildfil2)
        Button.__init__(self, master, image = self.baksida, command = self.vand)
        self.bild = self.baksida
        self.framsidaUpp = False

    def vand(self):
        if self.framsidaUpp:
            self.framsidaUpp = False
            self.configure(image = self.baksida)
            self.bild = self.baksida
            print("baksida")
        else:
            self.framsidaUpp = 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()
```

PROTOTYPER

Försök inte skriva allt på en gång!

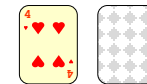
Arbeta med prototyper: körbara delversioner av programmet (kortspel0.py, kortspel1.py, kortspel2.py, etc)

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, Hjärtar och Spader)
 - framsidaUpp (True om kortet är uppvänt)
- Metoderna är
 - __init__ (konstruktorn),
 - __str__ (kortet som sträng),
 - vand (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.

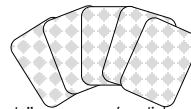
```
class Tal:  
    klassAttribut = 17  
  
    def __init__(self):  
        self.attribut = 42  
  
objekt = Tal()  
print(objekt.attribut)  
print(Tal.klassAttribut)
```

```
class Kort(object):  
    """ Ett spelkort. """  
    valörER = ["ess", "2", "3", "4", "5", "6", "7",  
              "8", "9", "10", "knekt", "dam", "kung"]  
    färgER = ["Klöver", "Ruter", "Hjärter", "Spader"]  
  
    def __init__(self, valör, färg, synligt = True):  
        """ Ett spelkort. """  
        pass  
  
    def __str__(self):  
        """ Visar kortets valör och färg om  
        framsidanUp, anars bara XX """  
        pass  
  
    def vänd(self):  
        """ Vänder kortet """  
        pass
```

```
class Kort(object):  
    """ Ett spelkort. """  
    VALÖRER = ["ess", "2", "3", "4", "5", "6", "7",  
              "8", "9", "10", "knekt", "dam", "kung"]  
    FÄRGER = ["Klöver", "Ruter", "Hjärter", "Spader"]  
  
    def __init__(self, valör, färg, synligt = True):  
        self.valör = valör  
        self.färg = färg  
        self.framsidanUp = synligt  
  
    def __str__(self):  
        if self.framsidanUp:  
            rep = self.färg + "-" + self.valör  
        else:  
            rep = "XX"  
        return rep  
  
    def vänd(self):  
        self.framsidanUp = not self.framsidanUp
```

HAND

- Klassen Hand representerar en korthand



- Klassens attribut är korten (en lista med kort).
- Metoderna är
 - `__init__`
 - `__str__`
 - `bort` (tar bort korten)
 - `stoppaIn` (lägger till ett kort)
 - `ge` (ger ett kort till en annan korthand)

HAND-OBJEKT

- Ett Hand-objekt har bara ett attribut: korten som är en lista med Kort-objekt.
- Rita ett exempel:

korten

valör	"4"	valör	"7"	valör	"8"	valör	"9"
färg	Hjärter	färg	Spader	färg	Hjärter	färg	Spader
framsidan	True	framsidan	True	framsidan	True	framsidan	True
0		1		2		3	

```
class Hand(object):

    def __init__(self):
        """ En korthand. Attributet "kortet" är en
            lista med Kort-objekt """

    def __str__(self):
        """ Returnerar sträng med alla korten """

    def bort(self):
        """ Tar bort alla kort i listan. """

    def stoppaIn(self, nyttKort):
        """ Stoppar in kortet "nyttKort" sist i
            listan """

    def ge(self, kort, annanHand):
        """ Plockar ut "kort" ur listan och stoppar
            in i ett annat Hand-objekt "annanHand" """
```

```
def __init__(self):
    """ Skapar en tom korthand. Attributet
        "kortet" är en
        lista med Kort-objekt """
    self.korten = []
```

```
def __str__(self):
    if self.korten:
        rep = ""
        for card in self.korten:
            rep += str(card) + "\t"
    else:
        rep = "<tom>"
    return rep.ljust(14)
```

```
def bort(self):
    self.korten = []
```

```
def stoppaIn(self, card):
    self.korten.append(card)
```

```
def ge(self, kort, annanHand):
    self.korten.remove(kort)
    annanHand.stoppaIn(kort)
```