Skolan för Datavetenskap och kommunikation

PROGRAMMERINGSTEKNIK

Föreläsning 9

IDAG:

- Klassen Bur
- Metoder som tar ett annat objekt som parameter
- Nästlade for-slingor

NYA HUVUDPROGRAMMET

```
n=int(input("Hur många djur vill du ha? "))

lista = skapaLista(n)

visa(lista)

slinga(lista)

adjö(lista)
```

Alla metoder använder `lista`. Vore det inte enklare att ha en klass med `lista` som attribut?

NY KLASS: BUR

```
class Bur(object):
# Flera virtuella husdjur i en bur

   def __init__(self, n):
   """Skapar en lista med n Husdjur"""
       self.lista = []
       for i in range(n):
          self.lista.append(Husdjur())
```

...FLER METODER I BUR

```
    def banna(self):
         for djur in self.lista:
            djur.banna()

    def mata(self, bullar):
         for djur in self.lista:
            djur.mata(bullar)

    def leka(self):
        for djur in self.lista:
            djur.leka()
```

KLASSEN HUSDJUR

Attribut	Metoder
namnet	__init__()
glad	namn()
hunger	bytaNamn()
kon	kontakt()
preferens	__str__()
	banna()
	mata()
	leka()
	avsked()

KLASSEN BUR

Attribut	Metoder
lista	__init__()
	banna()
	mata()
	leka()
	mingel()
	visa()
	avsked()

INTERAKTION MED ETT ANNAT OBJEKT

Hur kan man skriva en metod som använder två objekt?

Ha två parametrar: *self* och *other*

METOD-DEFINITION	ANROP
`def metod(self, other)`	`objekt1.metod(objekt2)`

KONTAKT

- Vi definierar en kontakt-metod som returnerar True om djur1 och djur2 får kontakt.
- Exempel: `if djur1.kontakt(djur2):`
- Tänk *`kontakt(djur1, djur2)`*

```
def  kontakt(self, kompis):
    """Testar om kontakt uppstår mellan
       detta husdjur (self) och kompis"""
    if (self.kön == kompis.kön):
        if (self.preferens == "samma") and\
           (kompis.preferens == "samma"):
           print("Puss!")
           return True
    elif (self.kön != kompis.kön):
        if (self.preferens == "annat") and\
           (kompis.preferens == "annat"):
           print("Puss!")
           return True
    else :
        return False
```

NÄSTLADE FOR-SLINGOR

Om man lägger en for-slinga inuti en annan säger vi att slingorna är nästlade. Den inre slingan går då igenom alla sina värden för varje värde i den yttre slingan. Exempel:

```
for i in range(3):
     for j in range(3):
          print(i,j)
```

```
0 0
0 1
0 2
1 0
1 1
1 2
2 0
2 1
2 2
```

Mingel

```
n = len(lista)
for i in range(n-1):
  jag = lista[i]
  for j in range(i+1,n):
    du = lista[j]
    if jag.kontakt(du):
     lista.append(Husdjur(jag.namn(),du.namn()))
```

djur 0 & djur 1	djur 0 & djur 2	djur 0 & djur 3
djur 1 & djur 2	djur 1 & djur 3	
djur 2 & djur 3		