

Skolan för Datavetenskap och kommunikation

DD1311 PROGRAMMERINGSTEKNIK MED PBL

Föreläsning 9

IDAG:

- ♥ Dela upp i funktioner
- ♥ Defaultparametrar
- ♥ Objekt som interagerar: meddelanden
- ♥ Egna moduler

DELA UPP I FUNKTIONER

- ♥ Hitta satser som hör ihop
- ♥ Vad ska funktionen heta?
- ♥ Vilka variabler ska vara parametrar?
- ♥ Vilka variabler ska skickas ut som returvärde?

DELA UPP I FUNKTIONER

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

#Skapar en lista med n stycken husdjur
lista = []
for i in range(n):
    nytt = Husdjur()
    lista.append(nytt)

#Visar alla husdjur
n = len(lista)
print("\n*** Du har", n, "husdjur: ***")
for djur in lista:
    print(djur)
print("*****\n")
```

FÖRSTA FUNKTIONEN

```
def skapaLista(n):
    #Skapar en lista med n stycken husdjur
    lista = []
    for i in range(n):
        nytt = Husdjur()
        lista.append(nytt)
    return lista
```

NYA HUVUDPROGRAMMET

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

lista = skapaLista(n)

visa(lista)

slinga(lista)

adjö(lista)
```

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):
    """Skäller på alla djur"""
    for djur in self.lista:
        djur.banna()

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

def leka(self):
    """Leker med varje djur"""
    for djur in self.lista:
        djur.leka()
```

MEDDELANDEN

♥ När ett objekt anropar en metod hos ett annat objekt så säger man att första objektet *skickar ett meddelande* till det andra objektet (och det andra objektet tar emot ett meddelande från det första objektet).

♥ Exempel: `djur1.kontakt(djur2)`

♥ Tänk `kontakt(djur1, djur2)`



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

DEFAULTPARAMETRAR

```
def __init__(self, namn1="", namn2=""):
```

♥ När en funktion/metod definieras kan man ge parametrar *defaultvärden*.

♥ Om parametern utelämnas vid anropet är det defaultvärdet som används. Exempel:

```
pappa = Husdjur()
barn = Husdjur("Conny", "Busse")
```

♥ Parametrar utan defaultvärde ska stå först i parameterlistan.

Speed dating

```
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		

KLASSEN HUSDJUR

Attribut	Metoder
namnet	<code>__init__()</code>
glad	<code>namn()</code>
hunger	<code>bytaNamn()</code>
kon	<code>kontakt()</code>
<code>__preferens</code>	<code>__str__()</code>
	<code>banna()</code>
	<code>mata()</code>
	<code>leka()</code>
	<code>avsked()</code>

KLASSEN BUR

Attribut	Metoder
lista	<code>__init__()</code>
	<code>banna()</code>
	<code>mata()</code>
	<code>leka()</code>
	<code>mingel()</code>
	<code>visa()</code>
	<code>avsked()</code>

LÄNK TILL HELA PROGRAMMET

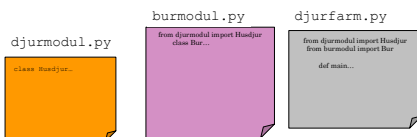
♥ husdjursfarm.py

MODULER



HUR GÖR MAN EN EGEN MODUL?

- ♥ Som ett vanligt Python-program utan main!
- ♥ Oftast en klass eller en samling funktioner. Exempel:



- ♥ OBS! När du gjort ändringar i modulen måste du starta nytt Shell-fönster i IDLE.