

DD1311 PROGRAMMERINGSTEKNIK MED PBL

Föreläsning 9

IDAG:

- ♥ Läs från fil
- ♥ Referenser
- ♥ Metoder som tar ett annat objekt som parameter
- ♥ Defaultparametrar
- ♥ Nästlade for-slingor

KLASSEN TENTA

Attribut	Metoder
<code>__bonus</code>	<code>__init__()</code>
<code>__poäng</code>	<code>__str__()</code>
<code>__totalpoäng</code>	<code>__lt__()</code>
<code>__betyg</code>	<code>visa()</code>
	<code>poängsumma()</code>
	<code>beräknaTotalpoäng()</code>
	<code>beräknaBetyg()</code>

LÄSA FRÅN FIL

Inläsningen måste anpassas till raderna i filen. Här står bonuspoängen först på raden, följt av "/", och därefter poängen på de åtta tentauppgifterna.

```
0 /9 11 3 0 12 5 0 8
5 /10 7 5 5 14 3 0 8
10 /8 12 4 6 16 4 4 8
8 /8 11 2 3 10 3 2 8
10 /10 14 6 7 20 10 6 8
10 /10 14 6 12 20 9 8 10
10 /10 12 6 8 14 5 4 6
0 /8 4 1 4 8 0 0 8
9 /10 10 5 0 4 2 5 9
10 /10 5 5 8 13 8 4 8
10 /10 14 5 6 20 8 6 7
```

```
tentorna = []
for rad in infile:
    lista = rad.split("/")
    bonus = int(lista[0])
    stränglista = lista[1]
    poänglista = str2int(stränglista)
    t = Tenta(bonus, poänglista)
    tentorna.append(t)
```

JÄMFÖRA MED ETT ANNAT OBJEKT

```
def __lt__(self, other):
    if self.__totalpoäng < other.__totalpoäng:
        return True
    else:
        return False
```

Om vi definierar metoden `__lt__` kan vi jämföra objekt med varandra.

Två parametrar: *self* och *other*, där *self* är det anropande objektet och *other* parametern.

Listans `sort()` anropar `__lt__` automatiskt!

SORTERA TENTORNA

```
print("Före sortering")
visaTentor(tentorna)
tentorna.sort()
print("\nEfter sortering")
visaTentor(tentorna)
```

REFERENSER

En listvariabel är en *referens* till listan, vilket innebär att tilldelning inte ger en kopia...

```

dagar = ["tisdag", "onsdag", "fredag"]
vecka = dagar
vecka[0] = "måndag"
print(vecka)
print(dagar)

```

vecka → måndag | onsdag | fredag
dagar → måndag | onsdag | fredag

VAD SKRIVS UT EFTER FÖLJANDE SATSER?

```

sorterat = tentorna
sorterat.sort()
visaTentor(tentorna)
visaTentor(sorterat)

```

- osorterat, sen sorterat efter totalpoäng
- osorterat, sen sorterat efter betyg
- 2 ggr sorterat efter totalpoäng
- 2 ggr sorterat efter betyg

KONTAKT



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

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.

KLASSEN HUSDJUR

Attribut	Metoder
<code>__namnet</code>	<code>__init__()</code>
<code>__glad</code>	<code>namn()</code>
<code>__hunger</code>	<code>bytaNamn()</code>
<code>__kon</code>	<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
<code>__lista</code>	<code>__init__()</code>
	<code>banna()</code>
	<code>mata()</code>
	<code>leka()</code>
	<code>mingel()</code>
	<code>visa()</code>
	<code>avsked()</code>

```
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		