



Skolan för Datavetenskap och kommunikation

Programmeringsteknik

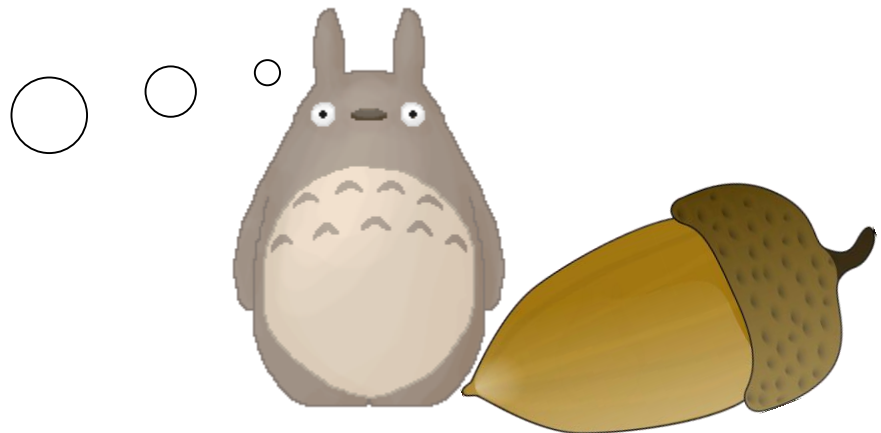
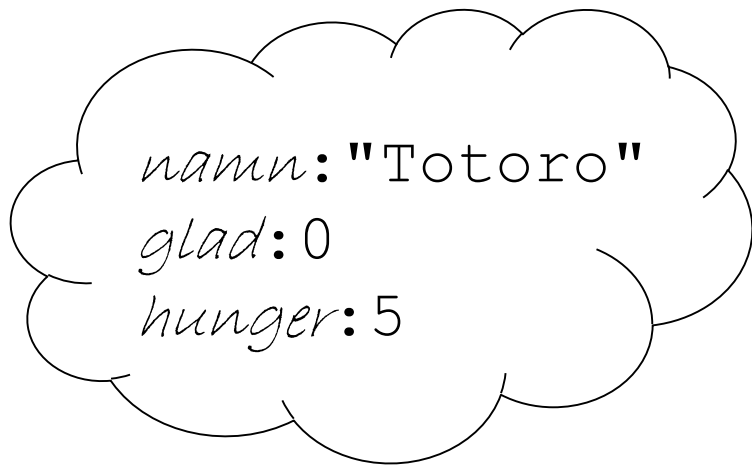
Föreläsning 6

Ny datatyp: objekt

- Kap 10 i kursboken
- Klasser (*classes*)
- Objekt (*objects*)
- Attribut (*attributes*)
- Metoder (*methods*)
- `__init__` och `__str__`

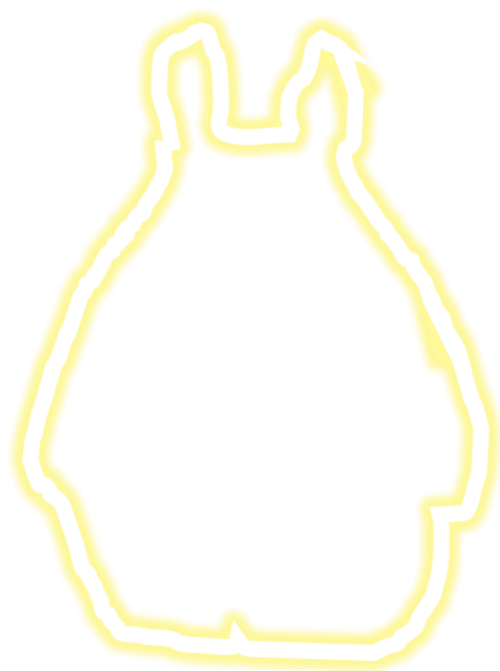
Objekt

- Objekt är en datatyp som kan innehålla både data och funktioner.
- Data som tillhör objektet kallas *attribut*.
- En funktion som hör till objektet kallas *metod*.



EXEMPEL: HUSDJUR

Klass	Attribut	Metoder
Husdjur	namn glad hunger	leka() banna() mata(ekollon) visaStatus()



Klasser

- Skriv en *klass* för dina objekt.
- Klassen fungerar som mall för objekten.
- I klassen beskrivs *attribut* och *metoder*.
- Alla metoder har parametern `self` som används för att nå attributen.

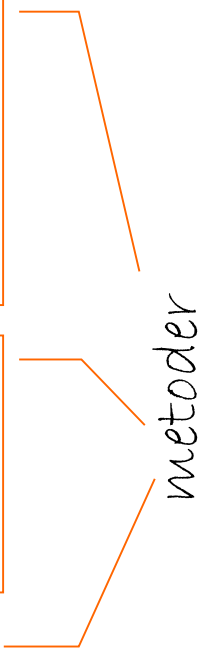
```
class Husdjur(object):  
    """ Ett virtuellt husdjur """
```

attribut

```
def __init__(self, djurnamn):  
    self.namn = djurnamn  
    self.glad = 0  
    self.hunger = 0
```

```
def banna(self):  
    """Ger husdjuret bannor."""  
    print "- Fy på dig", self.namn, "!"  
    self.glad -= 3
```

```
def mata(self, mat):  
    """Ger husdjuret mat."""  
    for i in range(mat):  
        print("GLUFS")  
    self.hunger -= mat
```



Hur skapa objekt ur klassen?

- I huvudprogrammet:

```
djur = Husdjur("Totoro")
```

- Metoden `__init__` anropas automatiskt
- Attributen ges värden
- En referens till objektet returneras

Metodanrop

- Metoder anropas alltid via objektet:

```
djur.leka()
```

```
djur.banna()
```

```
djur.mata(bullar)
```

```
djur.visaStatus()
```

- I övrigt fungerar metoder som vilken funktion som helst (kan ha parametrar och returvärdet)

klass, instans, self

- En *klass* är en generell beskrivning av vad objekten ska innehålla/kunna göra.
- Ett *objekt* är en *instans* av en klass. Variabeln `djur` refererar till ett objekt.
- I klassdefinitionen använder man *self* för att hänvisa till det anropande objektet.

```
djur.banna()
```

Metoden `__init__`

Ger attributen startvärden (**initiering**).

```
def __init__(self, djurnamn):  
    self.namn = djurnamn  
    self.glad = 0  
    self.hunger = 0
```

Metod: utskrift

```
def visaStatus(self):  
    """Visar husdjurets namn och tillstånd."""  
    print("\n", self.namn, "är", end = " ")  
    if self.glad > 0 and self.hunger < 3:  
        print("glad: (^_^)")  
    else:  
        print("ledsen: (T_T)")
```

Metoden `__str__`

Anropas automatiskt när man skriver ut objektet med `print`.

```
def __str__(self):  
    """Sträng som beskriver objektet"""  
    beskrivning = self.namnet  
    return beskrivning
```

```
def __str__(self):  
    """Alternativ variant"""  
    beskrivning = self.namnet + " är "  
    if self.glad > 5:  
        beskrivning += "glad: (^_^)"  
    else:  
        beskrivning += "ledsen: (T_T)"  
    if self.hunger > 3:  
        beskrivning += " och hungrig!"  
    else:  
        beskrivning += " och mätt."  
    return beskrivning
```

```
def main():
    djurnamn = input("Vad vill du kalla
                     ditt husdjur? ")
    djur = Husdjur(djurnamn)
    djur.visaStatus()
    svar = input(" Vill du \n
                 banna \n
                 mata \n \
                 leka med \n
                 ditt husdjur? " )
```


läxa 3

I labb 4 ska du skriva ett program med en egen klass.

Förbered dig så här:

- Läs Kap 10 i (om klasser) i kursboken.
- Gå igenom och provkör husdjursprogrammet så att du förstår precis hur det fungerar.
- Skriv sedan en egen klass (tillsammans med din gruppkompis) med minst fem attribut och fem metoder. Testa!