

## 2D1311 Programmeringsteknik med PBL

### Föreläsning 4

## Kapitel 7 & 8

- Filer
- Särfall (exceptions)
- Konservering (pickling)
- Klasser
- Objekt
- Attribut och metoder

## Filer

- Enda sättet att spara data mellan programkörningar är att lagra på fil.
- Det här behöver man kunna:
  - Öppna filen för läsning eller skrivning:  

```
infil = open("bok.txt", "r")
```
  - Läs in från en fil  

```
rad = infil.readline()
```
  - Skriva ut på en fil  

```
utfil.write("Natten var mörk.")
```
  - Stänga filen  

```
utfil.close()
```

"r" för läsning, "w" för skrivning

Fråga: Kan samma program både läsa från fil och skriva på fil?

## Konservering

- För att lagra en hel datastruktur (t ex en lista) på fil kan man *konservera* den:  

```
import cPickle
strumplista = ["röda", "blåa", "ylle"]
konserv = open("strumpor.txt", "w")
cPickle.dump(strumplista, konserv)
konserv.close()
```
- Den konserverade listan hämtas igen med metoden `load`

`konservera = pickle`



## Särfall

- När något blir fel i ett Python-program uppstår ett särfall, t ex `NameError`:

```
>>> print sko
```

Traceback (most recent call last):

```
File "<pyshell#17>", line 1, in -toplevel-
    print sko
```

`NameError: name 'sko' is not defined`

- Man kan ta hand om särfall genom att införa `try-except-else`-satser för de delar i programmet som kan krascha.

`särfall=exception`

Uppgift: Vilka fel har du fått när du kört Pythonprogram?

## Tipsfrågor:data i filen

- Data (tipsfrågorna) finns på filen `natur.txt`
- Varje frågeblock består av åtta rader:
  - Frågans rubrik: Fladdermusen
  - Frågan: Hur sover en fladdermus?
  - Fyra svarsalternativ: Bak-och-fram  
Ut-och-in  
Upp-och-ner  
Baklänges
  - Rätt svarsnummer: 3
  - Förklaring till svaret: Fladdermusen sover hängande i fötterna.

## Tipsfrågor:data i programmet

- De åtta raderna i ett frågeblock lagras i strängar (`category`, `question`, `correct`, `explanation`).
- Svarsalternativen läggs i en lista (`answers`).
- En heltalsvariabel för poängen (`score`).

## Tipsfrågor:algorithm

1. Öppna filen för läsning
2. Medan det finns frågor kvar i filen:
  - Läs in nästa fråga
  - Ställ frågan
  - Om svaret är rätt:
    - Ge beröm
    - Öka poängen med ett
3. Stäng filen

Funktion i tipsfrågor.py	Indata (parametrar)	Utdata (returvärde)
<code>open_file(file_name, mode)</code> #Öppnar filen	filens namn, öppningsmod	filen
<code>next_line(the_file)</code> #Läser in nästa rad	filen	nästa rad från filen
<code>next_block(the_file)</code> #Läser in nästa block	filen	nästa 8-block från filen
<code>welcome(title)</code> #Skriver ut titeln	titeln på frågesporten	-

```
# Michael Dawsons Trivia Challenge, kap 7,
# modifierad av Linda - 11 februari 2007

def open_file(file_name, mode):
    """Öppna en fil säkert för läsning eller utskrift."""
    try:
        the_file = open(file_name, mode)
    except (IOError), e:
        print "Kan inte öppna filen", file_name, "Avslutar\n", e
        raw_input("\n\nTryck retur för att stänga.")
        sys.exit()
    else:
        return the_file
```

```
def next_line(the_file):
    """Läs in och returnera nästa rad från filen."""
    line = the_file.readline()
    line = line.replace("/", "\n")
    return line
```

```
def next_block(the_file):
    """Returnera nästa datablock från filen."""
    category = next_line(the_file)

    question = next_line(the_file)

    answers = []
    for i in range(4):
        answers.append(next_line(the_file))

    correct = next_line(the_file)
    if correct:
        correct = correct[0]

    explanation = next_line(the_file)

    return category, question, answers, correct, explanation
```

```
def welcome(title):
    """Välkomnar spelaren och skriver ut frågesportstitel."""
    print "\t\tVälkommen till frågesporten!\n"
    print "\t\t", title, "\n"

def ask_a_question(question,answers):
    print question
    for i in range(4):
        print "\t", i + 1, "--", answers[i]

def check_answer(answer,correct,score,explanation):
    if answer == correct:
        print "\nJavisst!",
        score = score + 1
    else:
        print "\nFel.",
        print explanation
        print "Poäng:", score
    return score
```

Uppgift: Fyll i funktionstabellens  
sista tre rader!

```
def main():
    trivia_file = open_file("natur.txt", "r")
    title = next_line(trivia_file)
    welcome(title)
    score = 0
    category, question, answers, correct, explanation = \
        next_block(trivia_file)

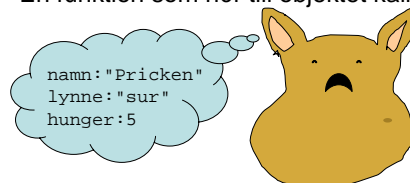
    while category:
        print category
        ask_a_question(question,answers)
        answer = raw_input("Vad svarar du? ")
        score = check_answer(answer,correct,score,explanation)
        category, question, answers, correct, explanation = \
            next_block(trivia_file)

    trivia_file.close()
    print "Det var sista frågan!"
    print "Din totalpoäng blev:", score

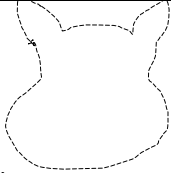
main() # Anropar main()-funktionen
```

## 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*.



# Klasser



- För att skapa objekt behöver man en *klass*.
- En klass är en mall för ett objekt
- I klassen beskrivs attribut och metoder.
- Alla metoder har parametern `self` som används för att komma åt de egna attributen.
- Metoden `__init__(self)` kallas *konstruktor* och anropas automatiskt då ett objekt skapas.

```
# Demonstrerar en husdjursklass (som Critter i kapitel 8)
import random

class Husdjur(object):
    """ Ett virtuellt husdjur """
    def __init__(self, namn):
        self.namn = namn
        self.lynne = random.choice(("sur", "arg", "ledsen"))
        self.hunger = 5
        print "Hej, jag heter", self.namn

    def visa_status(self):
        print "Just nu är jag", self.lynne,
        if self.hunger > 0:
            print "och hungrig!"
        else:
            print "och mätt!"

    def mata(self, mat):
        self.hunger -= mat

    def leka(self):
        self.lynne = random.choice(("glad", "sprallig", "busig"))
```

Labels in the image: 'attribut' points to `self.namn`, `self.lynne`, and `self.hunger`; 'konstruktor' points to `__init__`; 'metoder' points to `visa_status`, `mata`, and `leka`.

```
def main():
    pricken = Husdjur("Pricken")
    svar = raw_input("Vill du mata mig eller leka? ")
    if svar[0]=="m":
        pricken.mata(5)
    else:
        pricken.leka()
    pricken.visa_status()

main()
```

Uppgift: Om vi hade använt objekt för att representera frågeblocken i trivia-programmet - vilka attribut skulle objektet då haft?

## Klassattribut och statiska metoder

- Attribut och metoder hör alltid till ett visst objekt (Pricken är sur och hungrig)
- Men ett *klassattribut* är en variabel som är gemensam för alla objekt ur klassen.
- En variabel som skapas utanför metoderna i en klassdefinition blir automatiskt ett klassattribut.
- En *statisk metod* är en metod som arbetar på klassen. Då måste man ange att den är statisk:  
`welcome = staticmethod(welcome)`