

# Python

Objektorienterad programmering (och modellering)



Alumn – och  
examensmiddag VT20

# Dagens agenda

- Vad är objektorienterad programmering?
- Hur skiljer sig detta från funktionsdriven programmering? (det vi gjort hittills)
- Hur ser detta ut i Python?
- Lite snabba exempel!

# Informationsarkitekt

<http://edu.mah.se/sv/Program/TGIAA>



Läsår	LP1	LP2	LP3	LP4
1	Introduktion till datavetenskap	Introduktion till interaktionsdesign	Databasteknik	Informationsarkitektur I
	Introduktion till webbutveckling	Introduktion till programmering	Systemutveckling och projekt I	
2	Informationsarkitektur II	Informationsdesign	Metodik för mätning av användbarhet	
	Objektorienterad programmering och modellering för IA	Webbtjänster	Informationssäkerhet	Flerplattformsapplikationer med webbtekniker
3	Examensprojekt IA		Data- och informationsvetenskap: Examensarbete	
	Valbara kurser 30 hp			









00100101000101001000100101000101001000010100100001001010  
0101011010001010010101010110100010100101010101010101010  
10100010100110001000101000101001100010001010001010010001  
1010011010100001010010100110101010001010010001010010001  
000101010011010010000001010100110100100000101010  
010100010100100100100100010100100100100010100100010100  
01011001001001010101010110010010010010101010101010100  
0100111010101000100010011101010100010001001000100010011  
00100100101000100101001001001000101001000100100100010010  
010100001010010001001010000101001000100100010100100010010  
0110100010100101010101101001000101001010101010101010110  
00010100110001000101001001100110010001000100010100010001  
0011010100001010010100110011001000101000101001000101001  
1010100110100100000010101001101001000001010100000101010  
01000101001001001001000101001000101001000101001000101000  
110010010010101010101100100100100100101010101010101100  
0111010101000010001001110101010000100010001000100010011  
0100101000100100100100100100010100100010100100001010001  
00001010010000100101000010100001010010000101001000010100  
010001010010101010101101000101001000101001000101001010110  
010100110001000101000101001100110010001000100010100010001  
110101000010100101001100110101010000101001000101001010011  
101001101001000000101010011010010000001010100000101010001  
000101001001001001010001010001010010001010010001010010001  
100100100101010101011001001001001001001010101010101100  
0111010101000010001001110101010000100010001000100010011  
001001010001001001010010010010001001000100100010010001001  
0101000010100100001001000010100001010000101000010100001010  
10110100010100101010101101000101001000101001010101010110  
0100010100110001000101000101000101000101000100010100010001  
0100110101000010100101001101010100010100010100010100010011

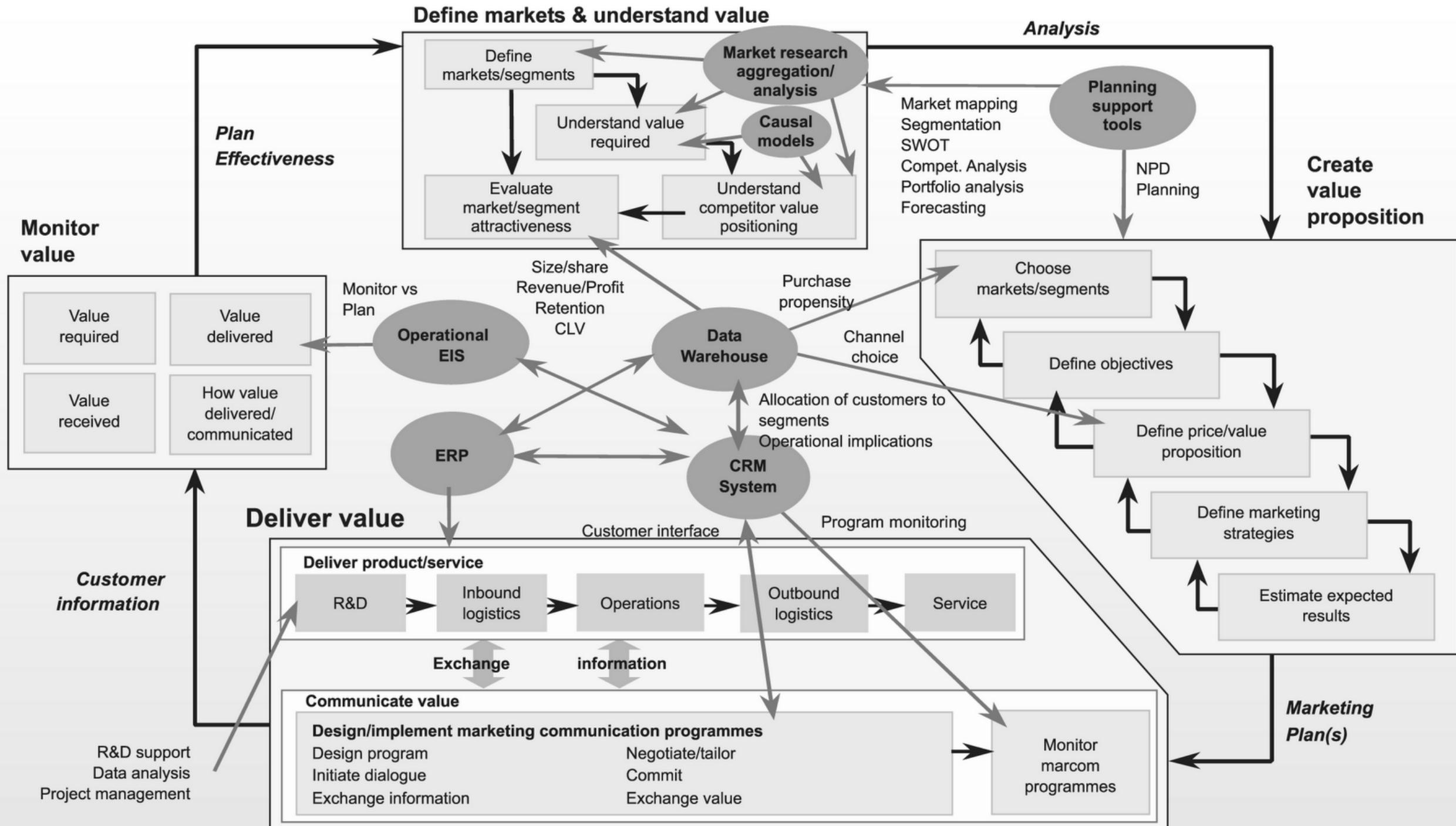


# Hur går ni till väga idag?

När ni jobbar med era projekt?









I Python-kod, beskriv en film?

I Python-kod, beskriv en film med  
skådespelare?

I Python-kod, beskriv en film med  
skådespelare, regissör, speltid, betyg,  
språk?

**THERE'S GOT TO BE A  
BETTER**

**WAY TO DO THAT**

memegenerator.net

# Datatyper?

# Vi vill modellera världen!

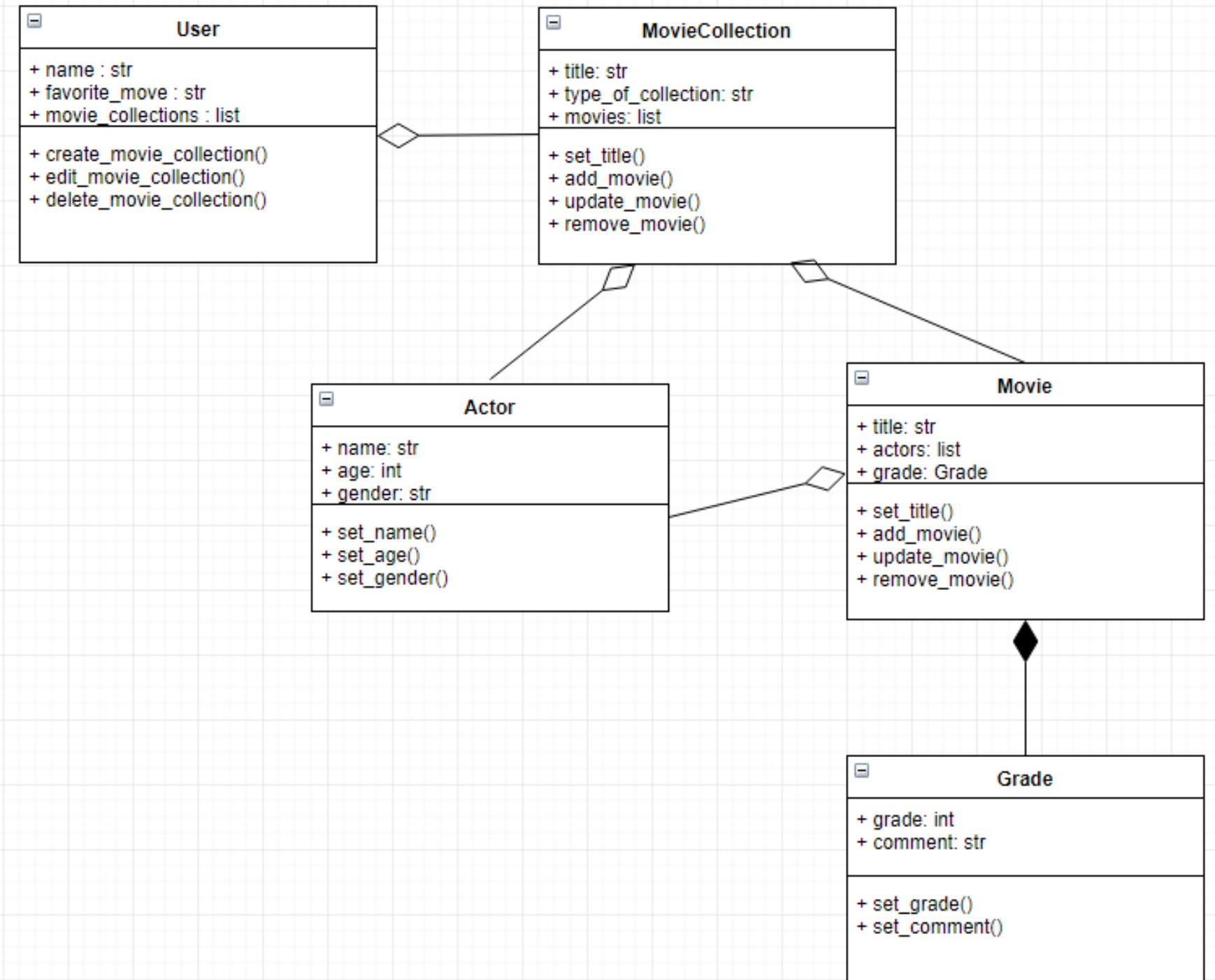
Objektorienterad programmering och modellering

# Vad är objektorienterad programmering

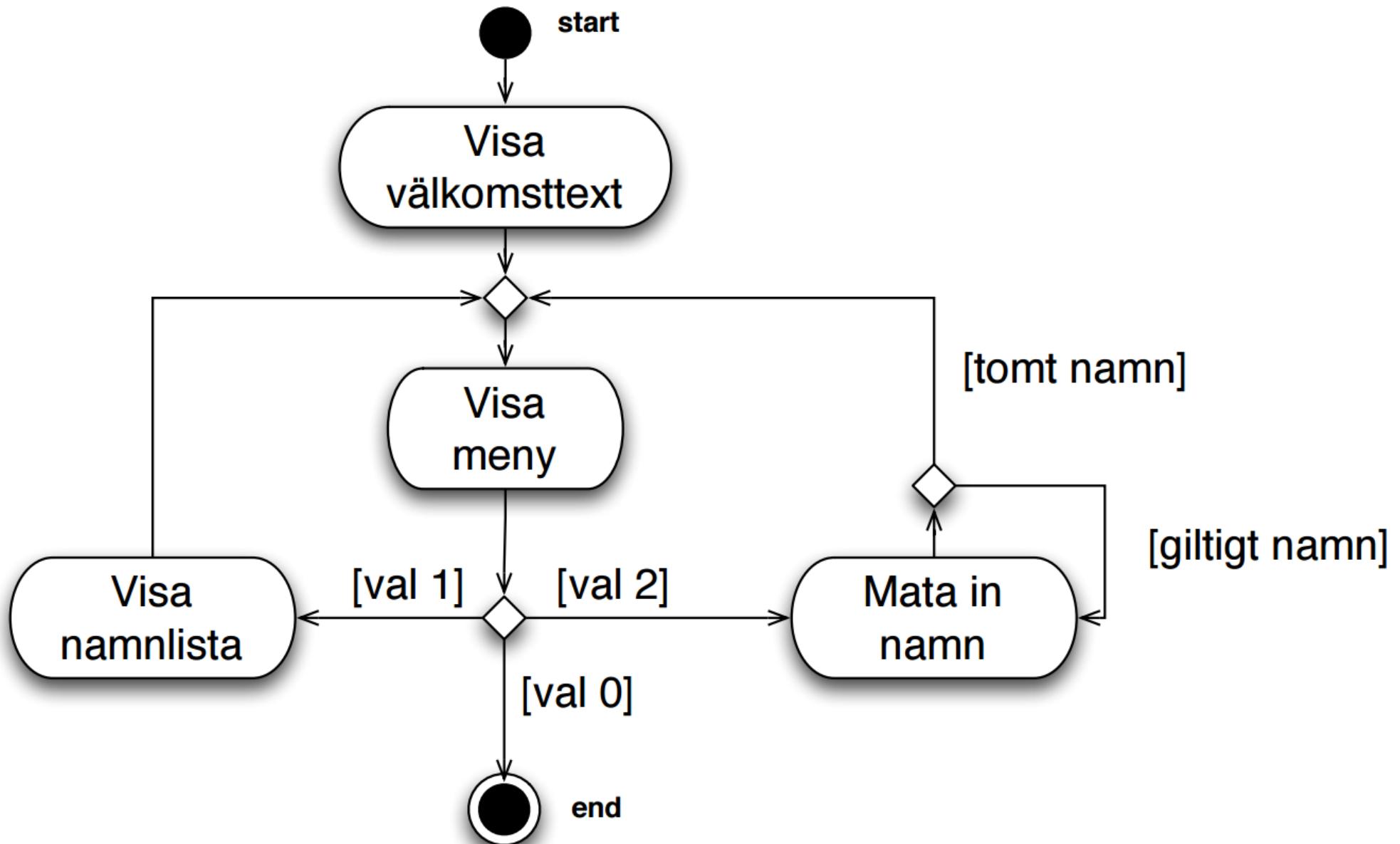
- En programmeringsparadigm, ett sätt hur man skriver & strukturerar kod.
- Vi vill kunna efterlikna den ”verkliga världen” så mycket som möjligt, genom att göra allt till objekt.
- Därför brukar man **modellera** sitt program innan man **programmerar** det.

# Att designa ett program

Varför inte en filmsamling?



**Funktionsdriven programmering** är en metod att skriva mjukvara. Den fokuserar på en mjukvarudesign som är centrerad på **funktioner** och **händelser** som sker i ett program.



# Funktionsdriven programmering består av en eller flera funktioner

Funktioner arbetar med data som är **fristående** från funktionen

Data skickas **mellan funktioner**

Fokus är på att bygga funktioner som hanterar en mjukvaras data

**Objektorienterad programmering** fokuserar på objekt. Objekt skapas från abstrakta datatyper och inkapslar data och funktioner tillsammans.

# Objektorienterad programmering

Ett **objekt** är en datatyp som innehåller både data och funktioner

Data som finns i ett objekt kallas för attribut (eller egenskaper)

Funktioner som finns i ett objekt kallas för **metoder**

# OBJECTS

# EVERYWHERE

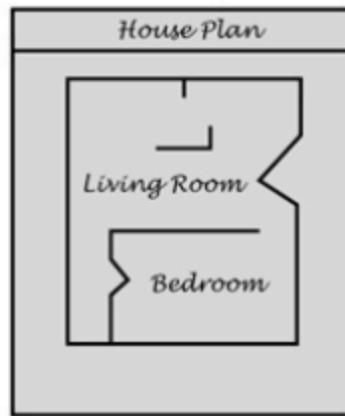


En **klass** är kod som specificerar en datatyp med **attribut** och **metoder**.

# Class

3 objects /  
instances /  
individuals

Blueprint that describes a house



Instances of the house described by the blueprint



Person

name

p1 : Person

name = “Jane”

p2 : Person

name = “John”

# Person

- name : str

---

+ get\_name : str

+ set\_name

+ say\_hello

+ \_\_str\_\_ : str

```
1 class Person(object):
2
3     def __init__(self, name):
4         self.name = name
5
6     def get_name(self):
7         return self.name
8
9     def set_name(self, name):
10        self.name = name
11
12    def say_hello(self):
13        print self.name, "says hello!"
14
15    def __str__(self):
16        return self.name
17
```

# Att modellera en stad

Tänk er typ "sim city"



RUMNEY

# Vad hittar vi för saker?

- Människa
  - *Attribut:*
  - Namn
  - Ålder
  - Kön
  - Pengar
  - *Metoder:*
  - Gå
  - Shoppa
  - Prata
  - Åka taxi
- 
- Byggnad
  - *Attribut:*
  - Gata
  - Nummer
  - Storlek
  - Våningar
  - Ålder
- 
- *Metoder:*
  - Bygga ut
  - Rasera
  - Renovera
- 
- Affär
  - *Attribut:*
  - Namn
  - Typ av affär
  - Adress
  - Telefonnummer
  - *Metoder:*
  - Öppna
  - Stänga
  - Sälja
  - Inventera



# Låt oss fokusera på taxibilen...

- Taxibil
- *Attribut:*
  - Förfare
  - Aktiv
  - Passagerare
  - Plats
- *Metoder:*
  - Hämta folk
  - Lämna folk
  - Byt chaufför
  - Byt passagerare

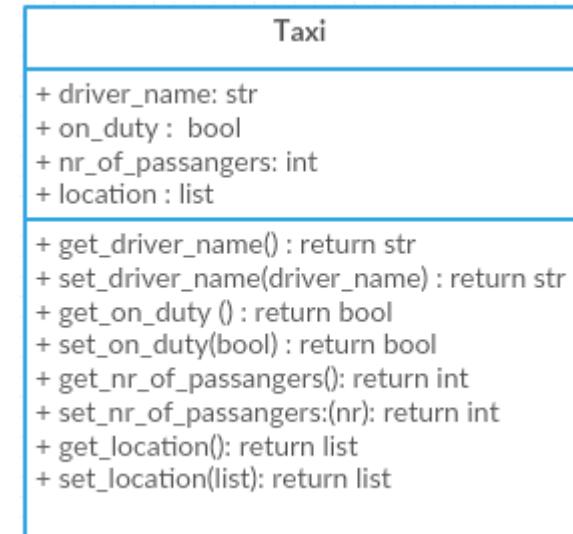


Vi har skapat datatypen:  
*Taxi*

# Vad är då en klass?

- En klass är en beskrivning av hur ett objekt (t.ex. en taxi-bil, en affär, en människa) ser ut. Alltså:
  - *Vilka attribut som finns*
  - *Vilka metoder som finns*
- Man brukar rita upp detta i ett så kallat klassdiagram.

- Taxibil
- *Attribut:*
  - Förare
  - Aktiv
  - Passagerare
  - Plats
- *Metoder:*
  - Hämta folk
  - Lämna folk
  - Byt chaufför
  - Byt passagerare



# Hur skiljer sig detta från det vi gjort hittills?

- Funktionsdriven programmering
  - *Skapar ett steg-för-steg program (funktioner som kallar på varandra i en viss ordning)*
  - *Vi är väldigt öppna med vilken data som finns – och hur denna skickas runt i programmet*
- Objektorienterad programmering
  - *Fokuserar på att modellera världen på ett sätt som är enkelt att förstå för människan*
  - *Istället för att man skickar data mellan olika funktionen har varje objekt sina egna funktioner och attribut – som ibland är hemliga, och objekten visar bara det som de vill visa.*

Vi kollar hur detta ser ut

# Men detta hade vi kunna göra innan ju!

```
def get_people():
    print "Picking up som people..."

def leave_people():
    print "Leaving som people..."

def print_taxis(taxi_car):
    if taxi_car["on_service"]:
        print "{0} is driving in {1} with {2} passangers".format(taxi_car["driver"], taxi_car["location"], taxi_car["nr_of_passengers"])
    else:
        print "{0} the taxi driver isn't working at the moment...".format(taxi_car["driver"])

taxi = {}
taxi["driver"] = "Anton"
taxi["on_service"] = True
taxi["nr_of_passengers"] = 2
taxi["location"] = "Lund" # Eller koordinater, long/lat
taxi["get_people"] = get_people
taxi["leave_people"] = leave_people

# Print taxi info
print_taxis(taxi)
# Picking up people
taxi["get_people"]()
# Ending service
taxi["on_service"] = False
# Print taxi info
print_taxis(taxi)
```

```
>>> ====== RESTART ======
>>>
Anton is driving in Lund with 2 passangers
Picking up som people...
Anton the taxi driver isn't working...
>>> |
```

