Seminar – Uke 7

Magnus Stokkeland (magsto@uio.no)

Plan for dagen

- Gjennomgang av pensum
  - Klasser/objekter
  - Interface/grensesnitt og innkapsling
  - Konstruktør
  - Livekoding av eksempel
- Trix-oppgaver

Objektorientert programmering

- Handler om å modellere virkeligheten ved å lage modeller som representerer egenskapene til noen eller noe
- En klasse beskriver et sett med objekter med samme oppførsel
- En klasse er ikke et objekt, men en modell som vi kan bruke til å lage objekter

```
class Klassenavn():

    def metodenavn(parameter):
        pass


objekt = Klassenavn()
```

Oppgave – skrive en klasse

Ut fra tegningen lagd av personene ovenfor lag en klasse person. Som har metodene:

- settNavn(self, navn) -> setter navnet på personen
- skrivNavn(self) -> skriver ut navnet å personen på dette formatet “Jeg heter <navn>”
- settAlder(self, alder) -> setter aldren på personen
- settVekt(self, vekt) -> setter vekten på personen
- settHoyde(self, hoyde) -> setter høyden på personen
- skrivUtInfo(self) -> som skriver ut infoen om personen
- skrivUtHilsen(self) -> som skriver ut en hilsen fra personen, på formen “Hei, jeg heter <navn> og jeg er <alder> aar gammel”
- hoyereEnn(self, annenPerson) -> tar inn en annen person og returnere True hvis personen er høyere enn den andre personen, False ellers

Grensesnitt og innkapsling

- Alle klasser har et «public interface» som beskriver hvilke tjenester klassen tilbyr
- Vi trenger ikke nødvendigvis vite hvordan en klasse er implementert (kodet) for å kunne bruke den
  - Eks: Bil.kjor(km) -> trenger vi lese koden for å skjønne hva denne metoden gjør?
- Vi vet ikke hvordan det skjer, vi bare vet at det skjer. Dette kalles innkapsling

Konstruktør

- En konstruktør bestemmer hva som må være med når vi lager et objekt av klassen
- F.eks, i forrige oppgave er det rart å opprette et objekt, og så måtte bruke «settNavn» hver gang
- Siden alle personer har et navn er det bedre å si at alle person-objekter MÅ ha et navn

```
class Klassenavn():
    def __init__(self, parameter):
        self._instansvariabel = parameter

    def metodenavn(parameter):
        pass
```

Oppgave – konstruktør

- Endre oppgave 1 slik at den nå i stedet for å ha metodene settNavn, settAlder, settVekt og setHoyde tar dette inn og setter disse (instans)variablene i konstruktøren istedenfor?

- Hvilke fordeler/ulemper har dette? Diskuter

Oppgave – opprette objekter

- Gitt klassen person som dere har laget over, opprett 3 instanser av klassen/objekter med navn/alder/vekt/hoyde du velger selv. Skriv deretter ut en hilsen fra hver person deres ved hjelp av skrivUtHilsen-metoden. Lagre deretter alle objektene i en liste.

- *Utfordring: se om du kan kalle på “skrivUtHilsen()”-metoden i en for-løkke*
- *Utfordring: lag en funksjon som tar inn en liste med personer, og returnerer den personen som er høyest*

Oppgave – dyr

- Skriv en klasse Dyr. Et dyr har følgende egenskaper:
  - Art
  - Kjønn
  - Vekt
- Lag en konstruktør (init-metode) og metoder for å hente ut data fra hver instansvariabel.
- Opprett tre objekter av klassen Dyr for å representere tre forskjellige dyr. Skriv ut informasjon om alle dyrene.