



GRUPPETIME
UKE 10

IN1000 GRUPPE 4 - 30.03.22



PLAN FOR GRUPPETIMEN

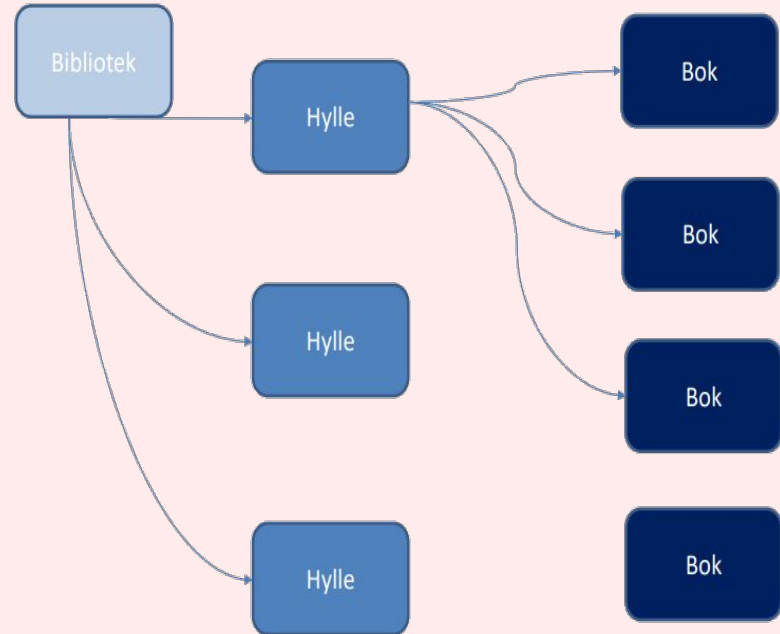
- Spørsmål angående oblig7?
- Aggregering
- Parprogrammering

LÆRINGSMÅL UKE 10

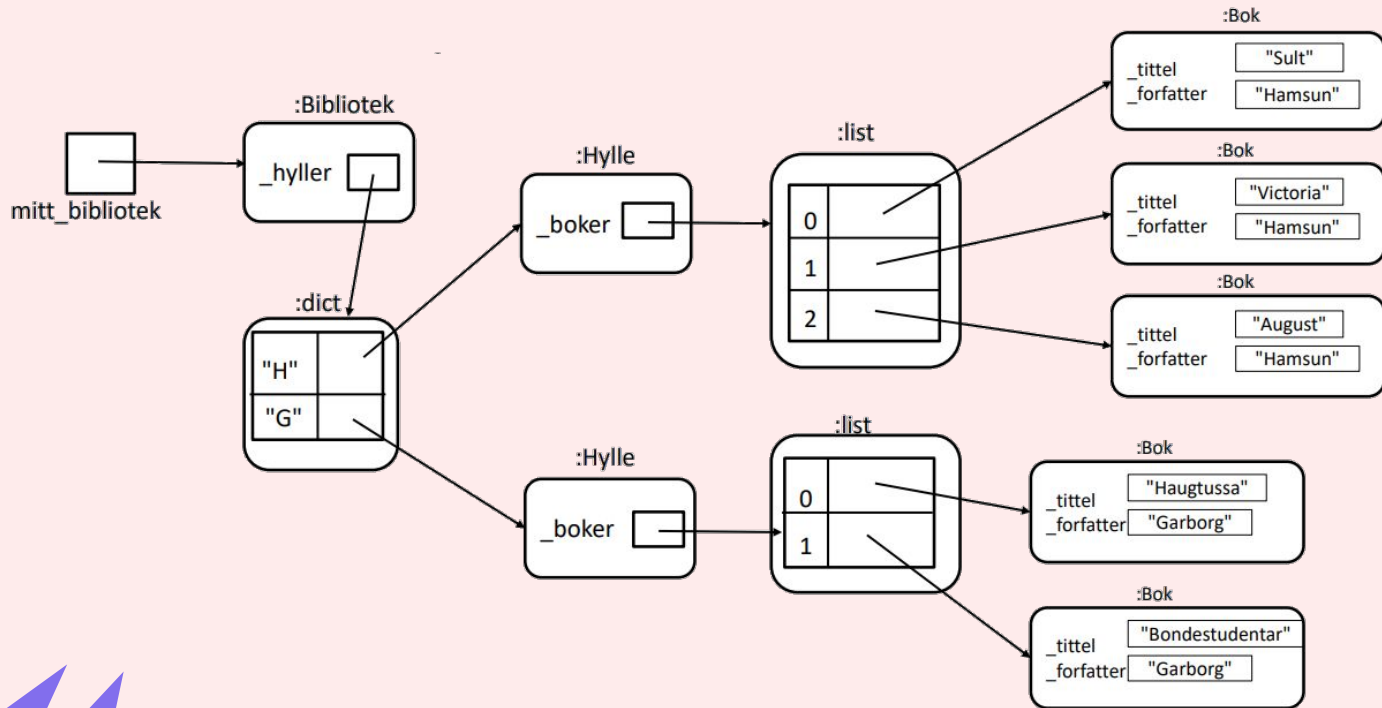
- Ha kjennskap til klassiske strukturer som er nyttige i programmering: lenkede lister, trær og grafer (ikke stress med denne, det er først og fremst aggregering dere skal kjenne til)
- Kunne praktisere aggregering
- **Det viktigste i denne og kommende uker er at dere trener på å skrive programmer med flere klasser, og bruker teorien som er gjennomgått for å forstå hvorfor og hvordan de virker som de gjør.**

AGGREGERING

- **Aggregering** kan beskrives som at et eller flere objekter av en klasse inngår som bestanddeler i et objekt av en annen klasse.
 - Et eksempel kan være en klasse Bibliotek der det finnes en liste eller ordbok som referer til objekter av en klasse Hylle. Vi kan ha flere ledd i en slik aggregert datastruktur: Klassen Hylle kan igjen referere til en samling objekter av klassen Bok.
 - Bibliotek → hyller → bøker



AGGREGERING: bokhylle eksempel



PARPROGRAMMERING

Denne uken skal dere begynne å jobbe på et større program

Dere skal lage klassene Fag og Student **som begge skal ha referanser til hverandre**

Neste uke vil dere jobbe med å knytte dem sammen til et større system

vi øver på aggregering!

```
class Fag:
    # 10-1.1
    def __init__(self, fagnavn):
        self._navn = fagnavn
        self._studentliste = []

    # 10-1.2
    def leggTilStudent(self, student):
        self._studentliste.append(student)

    # 10-1.3
    def hentAntallStudenter(self):
        return len(self._studentliste)

    #10-1.4
    def hentFagNavn(self):
        return self._navn

    # 10-1.5
    def skrivStudenterVedFag(self):
        print(self._navn)
        for student in self._studentliste:
            print(student.hentStudentNavn())
```

student.py

```
class Student:
    # 10-2.1
    def __init__(self, navn):
        self._navn = navn
        self._fagliste = []

    # 10-2.2
    def leggTilFag(self, fag):
        self._fagliste.append(fag)

    # 10-2.3
    def hentAntallFag(self):
        return len(self._fagliste)

    # 10-2.4
    def hentStudentNavn(self):
        return self._navn

    # 10-2.5
    def skrivFagPaaStudent(self):
        print(self._navn)
        for fag in self._fagliste:
            print(fag.hentFagNavn())
```


test_uke10.py

```
from fag import Fag
from student import Student
def hovedprogram():
    per = Student("Per")
    lisa = Student("Lisa")
    matte = Fag("MAT1001")
    in1000 = Fag("IN1000")

    # Per skal ta MAT1001
    per.leggTilFag(matte) #viktig at per legger til faget i sin oversikt/liste over fag
    matte.leggTilStudent(per) #og at faget legger til studenten i sin oversikt/liste over studenter.

    # Per skal også ta IN1000
    per.leggTilFag(in1000)
    in1000.leggTilStudent(per)

    # Lisa skal ta IN1000
    lisa.leggTilFag(in1000)
    in1000.leggTilStudent(lisa)

    # Utskrift av antall fag Per tar:
    per.skrivFagPaaStudent()
    print(f"Per tar {per.hentAntallFag()} fag\n")

    # Utskrift av antall fag Lisa tar:
    lisa.skrivFagPaaStudent()
    print(f"Lisa tar {lisa.hentAntallFag()} fag")

    # Utskrift av studenter som tar IN1000:
    in1000.skrivStudenterVedFag()

hovedprogram()
```

KONTAKT



...

sirisoll@uio.no

@sirisoll på Mattermost



UKENS MESTERVERK:

