

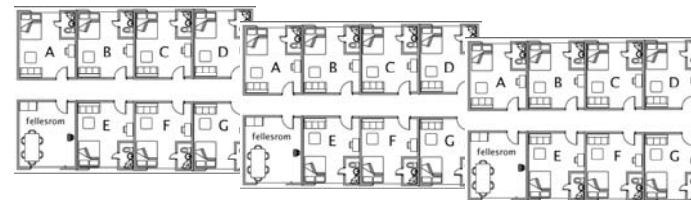
## Oblig 3 tips – litt mer tips enn i oppgaven

lørdag 18. okt 2008

Arne Maus

## Obligatorisk oppgave 3 Gulbrand Grås husleiesystem

I denne oppgaven skal vi se på Gulbrands tre-etasjes studenthus Utsyn. Hver etasje (1-3) har 7 hybler og et fellesrom. Hver hybel har et unikt hybelnavn som består av etasje + en bokstav (A-G). Da er 3E hybel E i 3. etasje, mens 2C er hybel C i 2. etasje. Nedenfor vises de 7 smakfulle hyblene i 3. etasje og fellesrommet. De 2 andre etasjene har tilsvarende hybler, bortsett fra at hyblene i 1. etasje leies umøblert og har derfor lavere husleie:



### Generell løsningsmetode

Husk vi skal lage en "kopi" av virkeligheten i Java-programmet

1. Finn **klassene** (hvilke objekter er det i problemet)
  1. Dataene som beskriver problemet (hvilke objekter har vi og hvor mange klasser er det ?)
2. Finn ut hvilke objekter som 'eier' objekter ( tegn **UML diagram**)
  1. Lag pekere (enkle pekere, arrayer eller HashMap'er) som binder systemet sammen
3. Finn ut **hva** systemet skal gjøre = metoder vi skal ha.
  1. Finn på et navn på hver av dem – foreløpig uten kode inni { }.
4. **Plasser metodene** (og andre datafelter) i de klassene de naturlig hører hjemme
5. Finn ut hvordan vi skal kjøre programmet – anta at vi har laget en klasse som er selve systemet – (class HusleieSystem)
  1. Lag et objekt av HusleieSystem (fra main)
  2. I konstruktoren leser du inn file(ne) i systemet og lager alle objektene som selve HusleieSystem 'eier' – for eksempel Et hybelhus 'Utsyn' med sine 21 hybler.
  3. Kall en 'kommandoløkke()' metode i HusleieSystem.
  4. Skriv ut nytt innhold av filer når 'kommandoløkke()' er ferdig
  5. Ferdig!
6. Lag et kjørende system med (bare) tomme metoder
7. Programmer hver metode for seg, test

3

### 1. Finn **klassene** - Substantiv-metoden : understrek alle

I likhet med de fleste andre universitetsbyer har Utopias hovedstad Amaurot dessverre stor mangel på studenthybler. Gulbrand Grå og hans firma HaiHus A/S lever av denne mangelen. Hybelhuset hans er så populær at du kan regne med at alle hyblene er opptatte nesten hele tiden og at han derfor også må holde orden på en venteliste av studenter som vil leie så fort en hybel blir ledig. Du skal hjelpe "Ærlige Gulbrand" å lage et system for å administrere utleie, utdeling av hyblene, og effektiv innkreving av husleie.

I denne oppgaven skal vi se på Gulbrands tre-etasjes studenthus Utsyn. Hver etasje (1-3) har 7 hybler og et fellesrom. Hver hybel har et unikt hybelnavn som består av etasjenr + en bokstav (A-G). Da er 3E hybel E i 3. etasje, mens 2C er hybel C i 2. etasje. Nedenfor vises de 7 smakfulle hyblene i 3. etasje og fellesrommet. De 2 andre etasjene har tilsvarende hybler, bortsett fra at hyblene i 1. etasje leies umøblert og har derfor lavere husleie:

univby, student, hovedstad, hybelhuset, hybel, venteliste, utleie, utdeling, innkreving, oppgave, etasje, fellesrom etasjenr, bokstav, husleie,..

4

# 1. Substantivene er forslag til klasser

**Alt** for mange

## A. Noen kutter vi ut

- A. Angår ikke datasystemet: universitetsby, hovedstad
- B. Noen er egentlig handlinger (utleie, utdeling, innkreving) og blir kanskje metoder

## B. Noen blir til enkle variable inne i en klasse

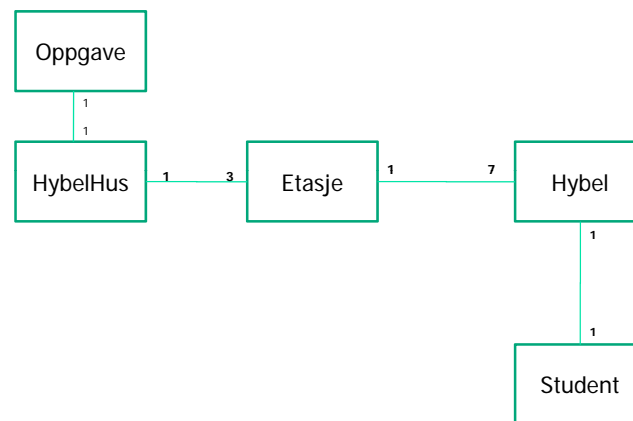
- A. De som ikke inneholder noe mer, ingen indre struktur (etasjenr, bokstav, husleie, ..)

## C. Noen blir til klasser

- A. Typisk de som er gjentatt *mange ganger* (hybel, hybelhus, etasje), men kanskje også bare én gang (student)

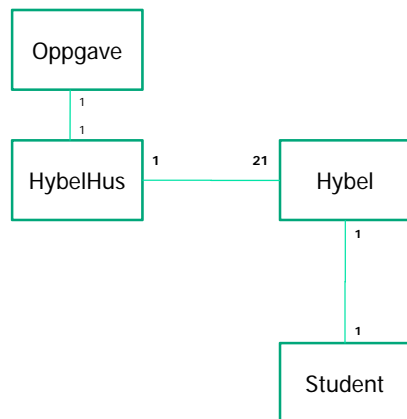
univby, student, hovedstad, hybelhuset, hybel, venteliste, utleie, utdeling, innkreving, oppgave, etasje, fellesrom, etasjenr, bokstav, husleie, ..

# 2. Tegn UML klassediagram (første forslag)2

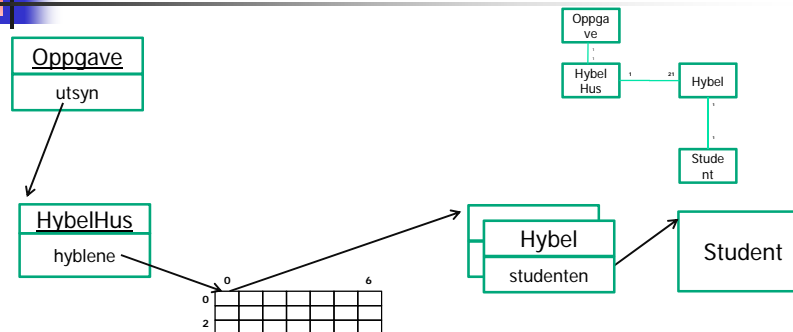


# 2. UML klassediagram (forslaget fra oppgave-teksten)

Hva er best ?



# 2b) Lag pekere ( nå henger det sammen)



### 3. Finn ut **hva** systemet skal gjøre = metoder vi skal ha

- Her er kravene til metoder fra oppgaven
  - alle (med ett mulig unntak) blir egne metoder
  - Tillegg: kommandoløkke() + meny()

1. Avslutt
2. Skriv oversiktstabell
3. Registrer ny leietager
4. Registrer utflytting av leietager
5. Registrer innflytting fra venteliste
6. Registrer betaling fra leietager
7. Måneds-kjøring av husleie
8. Sjekk om noen leietagere skal kastes ut

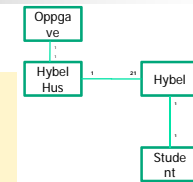
- Alle blir metoder – noen trenger parametre – det ordnes etterpå.

9

### 4. Plasser metodene (og andre datafelter) i de klassene de naturlig hører hjemme. Mange muligheter.

kommandoløkke() + meny()

1. Avslutt
2. Skriv oversiktstabell
3. Registrer ny leietager
4. Registrer utflytting av leietager
5. Registrer innflytting fra venteliste
6. Registrer betaling fra leietager
7. Måneds-kjøring av husleie
8. Sjekk om noen leietagere skal kastes ut



HybelHus: oversiktsTabell, innflyttingFraVenteliste, kommandoLøkke, meny, månedsKjøring, sjekkOmUtkasting

Hybel: regInnFlytting, regUtflytting, regBetaling, sjekkOmUtkasting

Student:

10

### 4. Plasser datafelter i klasser

- Det som ikke ble klasser
  - etasjenr, bokstav, husleie, venteliste,
- Student:
  - Navn, adresse,..?
- Andre variable
  - for eksempel når vi lager objekter (Student

11

### 5. Kjøre systemet

```
class HusleieSystem {
    public static void main (String[] args) {
        HybelHus hh = new HybelHus();
        hh.kommandoLøkke();
    }
}
```

12

## "hybeldata.txt" – systemets database

Først står det en linje for hver hybel, 21 linjer i alt, med hver linje på følgende form:

```
int etasje; char bokstav; String studentnavn; int saldo;
```

Et eksempel på en linje kan være:

```
2; C; Ole Brun; 2400;
```

For tomme hybler settes studentnavn lik "TOM HYBEL" og 0 i siste tall. Etter disse 21 linjene skal det stå en siste linje med format:

```
int måned; int år; int totaltAntallMåneder; int totalFortjeneste; 0;
```

13

## Lesing av redusert "hybeldata.txt":

```
2; C; Ole Brun; 2400;
3; A; TOM HYBEL; 3000;
```

Når vi har med en string 'sep' av separatortegn til en innlesningsmetode i klassen In, er det som leses en gruppe med tegn som er de tegnene som finnes mellom to skilletegnene i teksten sep

For å si det enklere:

Først hoppes det over alle skilletegn i 'sep', så leses alt frem til neste skilletegn som det vi egentlig er interessert i.

Eks. `int i =fil.inInt(" ;");`

vil først hoppe over alle blanke og ; , så lese de neste tegnene til første 'blanke' eller ; og prøve å tolke det som et heltall.

14

```
1 import easyIO.*;
2
3 class Student {
4     String navn;
5     int saldo;
6     Student(String navn, int saldo) {
7         this.navn = navn;
8         this.saldo = saldo;
9     }
10    void skrivData() {
11        System.out.println("Student:" + navn + ", med saldo:" + saldo);
12    }
13 }
14
15 // 2; C; Ole Brun; 2400;
16 // 3; A; TOM HYBEL; 3000;
17
18 public class LesHybelData {
19     public static void main(String [] args) {
20         int i = 0;
21         int antLinjer = 2;
22
23
24         In hdata= new In("hybeldata.txt");
25
26         while (i++ < antLinjer ) {
27             int et = hdata.inInt("; ");
28             char hbokstav = hdata.inChar("; ");
29             String navn = hdata.inWord("; ");
30             int saldo = hdata.inInt("; ");
31             Student s = new Student(navn,saldo);
32             s.skrivData();
33         }
34
35         hdata.close();
36     }
37 }
```

M:\INF1000\Oblig3tips>java LesHybelData  
Student:Ole Brun, med saldo:2400  
Student:TOM HYBEL, med saldo:3000

Merk ingen blanke skilletegn når vi leser inn navnet - HVORFOR

15

## Tabell

### 1. Skriv oversiktstabell:

Denne metoden skal skrive ut en oversikt som viser, for hver hybel: hybelnavn, studentnavn (eller "ledig" hvis hybelen er ledig), og saldo (eller skyldig beløp hvis det er i minus). Utskriften kan formateres i en tabell som vist under, eller på en annen måte.

| Hybel         | Leietager        | Saldo |
|---------------|------------------|-------|
| 1A            | ( ledig )        | 0     |
| 1B            | Albert Johnsen   | 4300  |
| 1C            | Johanne Ullersen | 9000  |
| 1D            | ( ledig )        | 0     |
| .... osv .... |                  |       |

Måned: 10.2008. Måneder i drift: 48. Totalfortjeneste: 274000.

Til slutt skrives det ut nåværende måned, antall måneder systemet har vært i drift, og total fortjeneste for Gulbrand.

16



## Tips 3

3. I skjelettet av løsningen ovenfor er det en Student-peker leietager i Hybel-klassen. For å teste om den peker på et objekt, kan man teste:

```
if (leietager == null)
```

`null` er et Java-ord som er pekerverdien 'intet objekt'. Dvs. if-testen slår til hvis `leietager` ikke peker på et objekt. `null` kan også brukes i tilordningssetninger for eksempel hvis `leietager` peker på et objekt, kan vi få fjernet det med: `leietager = null;`

17



## Tips 4

4. Vi har behov for å plassere alle Hybel-objektene i en array. I program-eksempelet er det nyttig en todimensjonal array, hvor første indeksen går på etasjennummer og andre på hybel innen i en etasje. Nå er problemet at hybelen har et bokstavnavn (A,B,...,H)og ikke et tall. Dette kan løses på følgende måte:


```
System.out.print("Gi etasje:");
int etg = tast.inInt();
System.out.print("Gi hybelbokstav:");
tast.skipWhite();
bokstav = tast.inChar();
i = (int) (bokstav - 'A');
```

Da vil: `hyblene[etg-1][i]` finne riktig peker i arrayen. Og motsvarende, hvis vi vet numeret 'i' til en hybel i etasjen , og vil ha bokstaven , gjør vi det slik:

```
char b = (char) ('A' + i);
```

Begge disse omgjøringene fra char til heltall og motsatt, hviler på at bokstavene A,B,... ligger etter hverandre i kodingen av tegnsettet.

18



## Tips 5 Out xx = new Out("venteliste.txt",true);

5. Når du skal skrive på en fil flere ganger, som ved utskrift til "`venteliste.txt`" i deloppgave 2, og ønsker å legge noe *til* på slutten av filen (ikke overskrive det som står der allerede), må du åpne fila med *append* – dvs. at når du utfører instruksjonen *new*, så har du med en parameter nr. 2 som er `true`. Det opprettes da en ny fil hvis filnavnet er nytt, men hvis ikke, åpnes den gamle fila på slutten av innholdet den hadde fra før. F.eks. `= new Out("venteliste.txt", true).`

19



## Tips 7

```
if (new File("hybeldata.txt").exists() ) {
    // Kode for å lese fila:
    ... = new In("hybeldata.txt");
    // Les data og opprett objektene som beskrevet på fila...
    ...close();
} else {
    // Her kommer programmet hvis datafilen ikke finnes fra før.
    // Du må da opprette alle objektene som representerer
    // et hybelhus uten beboere
}
```

20



## Tips 8:

---

8. For å sikre at den månedlige beregningen av husleie og fortjeneste bare skjer en gang per måned, må du nok både spørre Gulbrand om hvilken måned han skal ha husleiekraft for, samt på fila "hybeldata.txt" ha opplysninger om hvilke måneder det allerede er skrevet ut husleie for.



---

Lykke til !