

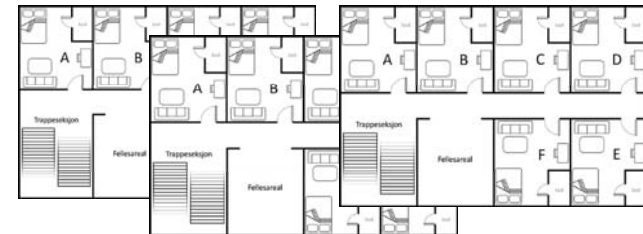
## Oblig 3 Hybelhus – litt mer tips enn i oppgaven

lørdag 15. okt 2011

Arne Maus

## Obligatorisk oppgave 3 Gulbrand Grås husleiesystem

I denne oppgaven skal vi se på hans studenthus Utsyn. Utsyn består av 3 etasjer, nummerert fra 1-3. I hver etasje finner vi 6 hybler og et fellesareal. Hver hybel navngis med etasje + en bokstav. Første hybel har bokstaven A, siste hybel har bokstaven E. Fellesarealet er ikke markert med bokstav. For eksempel 2A en hybel i 2. etg. i andre etasje, mens 1E er hybel E i første etasje.



2

## Inntekter og utgifter for Gulbrand

- Inntekter: Gulbrand Grå leier ut hyblene i de to nederste etasjene for 6000 kroner i måneden, mens husleien for en hybel i toppetasjen er 7000 kroner på grunn av utsikten. Beboerne betaler husleie for sine hyblene til Gulbrand – og husleien dekker alle utgifter på hybelen unntatt strøm, som beboerne betaler utenom dette systemet.
- Utgifter: Gulbrand har hyret inn firmaet Vedlikehold A/S som tar seg av alle løpende utgifter: Vedlikehold, reparasjoner, kommunale avgifter, nettabonnement i hyblene, og utstyr og strøm til fellesarealer. For dette betaler Gulbrand hver måned til Vedlikehold A/S: 1400 kr per hybel A-F i hver etasje, uansett om en hybel har beboer eller ikke, pluss 2100 kr per etasje for fellesarealer.

3

## 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 eller arrayer ) 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

4

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

Gulbrand Grå har et hybelhus kalt Utsyn, med 18 studenthybler som han leier ut til studenter i Oslo. Du skal lage et system for å administrere utleie av hyblene i hybelhuset. Utsyn har 3 etasjer, nummerert fra 1 til 3. I hver etasje er det 6 hybler, kalt rom A til F, og et fellesrom. Hver hybel har et entydig «hybelnavn» som består av etasjenummer og rom-bokstav, f.eks. heter hybelen i rom C i andre etasje "2C". Nedenfor ser du en illustrasjon av en etasje. Alle etasjene har samme planløsning.

**Inntekter:** Gulbrand Grå leier ut hyblene i de to nederste etasjene for 6000 kroner i måneden, mens husleien for en hybel i toppetasjen er 7000 kroner på grunn av utsikten. Beboerne betaler husleie for sine hyblene til Gulbrand – og husleien dekker alle utgifter på hybelen unntatt strøm, som beboerne betaler utenom dette systemet.

**Utgifter:** Gulbrand har hyret inn firmaet Vedlikehold A/S som tar seg av alle løpende utgifter: Vedlikehold, reparasjoner, kommunale avgifter, nettabonnement i hyblene, og utstyr og strøm til fellesarealer. For dette betaler Gulbrand hver måned til Vedlikehold A/S: 1400 kr per hybel A-F i hver etasje, uansett om en hybel har beboer eller ikke, pluss 2100 kr per etasje for fellesarealer.

5

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

Gulbrand Grå har et hybelhus kalt Utsyn, med 18 studenthybler som han leier ut til studenter i Oslo. Du skal lage et system for å administrere utleie av hyblene i hybelhuset. Utsyn har 3 etasjer, nummerert fra 1 til 3. I hver etasje er det 6 hybler, kalt rom A til F, og et fellesrom. Hver hybel har et entydig «hybelnavn» som består av etasjenummer og rom-bokstav, f.eks. heter hybelen i rom C i andre etasje "2C". .....

**Inntekter:** Gulbrand Grå leier ut hyblene i de to nederste etasjene for 6000 kroner i måneden, mens husleien for en hybel i toppetasjen er 7000 kroner på grunn av utsikten. Beboerne betaler husleie for sine hyblene til Gulbrand – og husleien dekker alle utgifter på hybelen unntatt strøm, som beboerne betaler utenom dette systemet.

**Utgifter:** Gulbrand har hyret inn firmaet Vedlikehold A/S som tar seg av alle løpende utgifter: Vedlikehold, reparasjoner, kommunale avgifter, nettabonnement i hyblene, og utstyr og strøm til fellesarealer. For dette betaler Gulbrand hver måned til Vedlikehold A/S: 1400 kr per hybel A-F i hver etasje, uansett om en hybel har beboer eller ikke, pluss 2100 kr per etasje for fellesarealer.

6

## 1. Substantivene er *forslag* til klasser

**Alt** for mange

### A. Noen **kutter vi ut**

- A. **Angår ikke datasystemet:** Vedlikehold, reparasjoner, kommunale avgifter, nettabonnement, strøm, utsikt,..
- B. **Noen er egentlig handlinger** (utleie, utdeling, innkreving, ) og blir kanskje metoder

### B. Noen er egentlig **det samme** og blir til ett forslag

- A. (beboer, student)

### C. Noen blir til **enkle variable** inne i en klasse

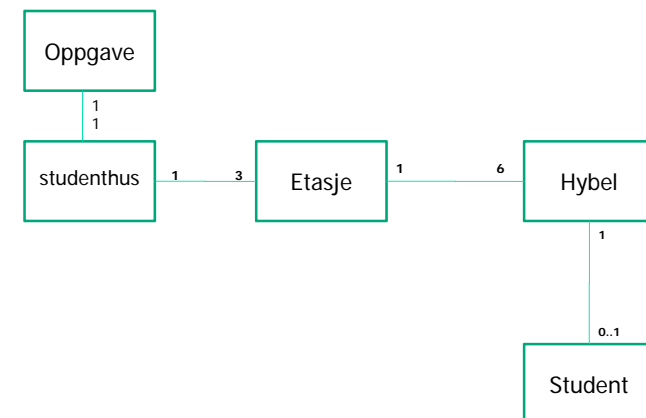
- A. De som ikke inneholder noe mer, ingen indre struktur (etasjenr, bokstav, husleie, måned ..)

### D. Noen blir til **klasser**

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

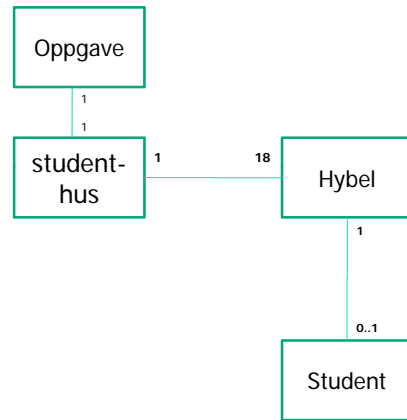
7

## 2. Tegn UML klassediagram (første forslag)



8

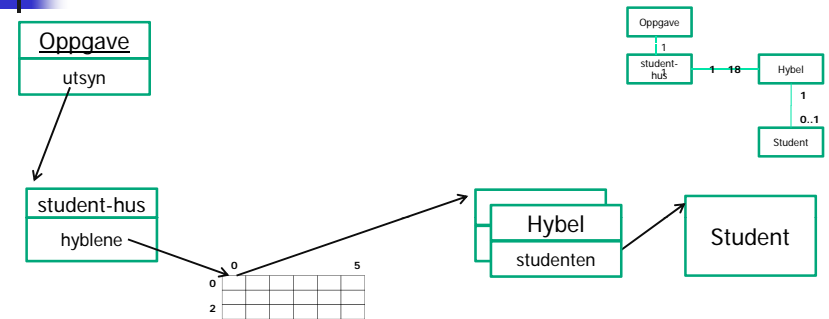
## 2. UML klassediagram (forslaget fra oppgave-teksten)



Hva er best ?

9

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



10

## 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()

- Skriv oversikt
- Registrer ny leietager
- Registrer betaling fra leietager
- Registrer frivillig utflytting
- Månedskjøring av husleie
- Kast ut leietagere
- Avslutt

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

11

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

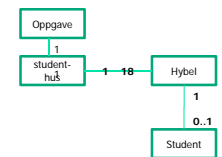
kommandoløkke() + meny()

- Skriv oversikt
- Registrer ny leietager
- Registrer betaling fra leietager
- Registrer frivillig utflytting
- Månedskjøring av husleie
- Kast ut leietagere
- Avslutt

HybelHus: oversiktsTabell, kommandoLøkke, meny, månedskjøring, sjekkOmUtkasting

Hybel: regInnFlytting, regUtflytting, regBetaling, sjekkOmUtkasting

Student:



12

## 4. Plasser datafelter i klasser

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

13

## 5. Kjøre systemet

```
class Oppgave {  
    public static void main (String[] args) {  
        StudentHus gg = new StudentHus ();  
        gg.kommandoLøkke();  
    }  
}
```

14

## "hybeldata.txt" – systemets database

### Datafilen "hybeldata.txt"

Første linje i filen inneholder fire heltalls-verdier adskilt med semikolon:

```
int måned; int år; int totalfortjeneste; int totaltAntallMåneder.
```

Her er "int måned" og "int år" månedsnummeret og årstallet da månedskjøring sist ble utført, hvor måned er et tall i området 1 til 12, og årstallet er firesifret. Det tredje tallet er Gulbrands totale fortjeneste siden systemet ble satt i drift; og det siste tallet angir antall måneder systemet har vært i drift.

Deretter er det **18 linjer**, en for hver hybel, med følgende format:

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

For tomme hybler skal studentnavnet lagres i datafilen som TOM HYBEL, med saldo 0. Du kan anta at alle studenter har unike navn. Alle disse dataene skal holdes oppdatert internt i programmet mens det kjører, og skal skrives tilbake til datafilen når programmet avsluttes. Slik kan Gulbrand starte og avslutte programmet uten å miste data mellom kjøringene.

15

## Lesing av en redusert "hybeldata.txt":

```
2; C; Ole Brun; 2400;  
3; A; TOM HYBEL; 0;
```

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.

16

```

import easyIO.*;
class Student {
    String navn;
    int saldo;
    Student(String navn, int saldo) {
        this.navn = navn;
        this.saldo = saldo;
    }
    void skrivData(char hybel,int et ) {
        System.out.println("Student:" + navn +", med saldo:" + saldo
            +", etasje:" +et+", hybel:" +hybel );
    }
}

// 2: C: Ole Brun; 2400;
// 3: A: TOM HYBEL; 0;

public class LesHybelData {
    public static void main(String [] args) {
        int i = 0;
        int antLinjer = 2;

        In hdata= new In("hybeldata.txt");

        while (i++ < antLinjer ) {
            int et = hdata.readInt("; ");
            char hbokstav = hdata.inChar("; ");
            String navn = hdata.inWord("; ");
            int saldo = hdata.readInt("; ");
            Student s = new Student(navn,saldo);
            s.skrivData(hbokstav, et);
        }

        hdata.close();
    }
}

```

Z:\INF1000\Obl3tips>java LesHybelData  
Student:Ole Brun, med saldo:2400, etasje:2, hybel:C  
Student:TOM HYBEL, med saldo:0, etasje:3, hybel:A

Merk ingen blanke skilletegn når vi leser inn navnet - HVORFOR

# 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.



## Tips 2

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;



## Tips 3

Hver gang metoden **tilkallTorpedo()** blir kalt skal du skrive til slutten av filen **torpedo.txt** (uten å overskrive det som allerede ligger der). Da må du åpne filen med *append*.

Det gjøres ved å ha en ekstra parameter når du åpner filen:

```
Out fil = new Out("Torpedo.txt", true);
```

Husk alltid å lukke filer når du er ferdig å lese/skrive til dem!



## Tips 4

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

```
skjerm.out("Oppgi hybelnavn, etg bokstav: ");
int etg = (int) (tast.inChar(" ") - '1'); // '1' gir 0, '2' gir 1, osv.
char bokstav = tast.inChar(" "); // Les rombokstaven
int rom = (int) (bokstav - 'A'); // 'A' gir [0], 'B' gir [1], osv.
```

Da vi `Hyblene[etg][rom]` finne riktig peker i arrayen. Og motsvarende, hvis vi vet nummeret '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.

21



## Tips 5

```
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
}
```

22



## Tips 6:

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 husleiekrav for, samt på fila "hybeldata.txt" ha opplysninger om hvilke måneder det allerede er skrevet ut husleie for.

23

```
import easyIO.*;

class Oppgave {
    public static void main(String [] args) {
        Utsyn s = new Utsyn();
        s.menyLøkke();
    }
}

class Student {
    String navn; // studentens navn
    int saldo; // studentens saldo

    // evt. metoder for å behandle studentene
}

class Hybel {
    Student leietager; // peker på leietager. Er hybelen tom, er denne null
    boolean toppEtasje; // true = 7000kr/mnd i leie
                    // false = 6000kr/mnd i leie

    // evt. metoder for å behandle hyblene?
}
```

24

```

class Utsyn {
    Hybel[][] hyblene = new Hybel[3][6];

    // deklarer objektvariable for
    // økonomi-/månedsberegning

    In tast = new In();
    Out skjerm = new Out();
    String DATAFIL = "Hybeldata.txt";

    // Konstruktør for klassen Utsyn
    Utsyn() {
        // opprett hybler:
        // hyblene[etg][rom] = new Hybel();
        // Les inn fra datafil. Opprett studentobjekter
        // for de hyblene hvor det bor studenter:
        // hyblene[etg][rom].leietager =
        //         new Student();
    }
}

```

```

void menyLøkke() {
    int kommando = -1;
    while (kommando != 7) {
        // skriv ut menyen
        skjerm.outln("Hovedmeny");
        skjerm.outln("7. Avslutt");

        // hent kommando fra bruker
        skjerm.out("Kommando: ");
        kommando = tast.inInt();

        // utfør kommando
        switch(kommando) {
            case 1: skrivOversikt(); break;
            case 2: regNyLeietager(); break;
            case 3 : regBetaling(); break;
            case 4: regFrivilligFlytting(); break;
            case 5:
                // ... fyll ut resten selv
                default: // gi feilmelding
            }
        }
    }
}

```

25

```

// metoder for de forskjellige kommandoene i menyLøkke()

void skrivOversikt() { /* ... */ }
void regNyLeietager() { /* ... */ }
void regFrivilligFlytting() { /* ... */ }
void regBetaling() { /* ... */ }

// ... fyll ut med minst 4 ekstra metoder

void avslutt() { /* ... */ }
}

```

26



Lykke til !