Anbefalt fremgangsmåte – og noen tips

1. Bestem datastrukturen for data om hybelhuset – tenk objektorientert! Du finner tips om valg av klasser blant annet i kapittel 8.3 i læreboken.
2. Lag et skjelett for programmet ditt etter den malen dere har lært – én klasse med main-metoden, deretter den eller de klassen(e) du har bestemt deg for i punkt 1. I første omgang holder det med klassenavn, evt kommentar om planlagt innhold.
3. Hvilke variabler trenger du i de forskjellige klassene (f.eks. for navn, saldo, fortjeneste, osv.)?, og hvilke av disse bør være arrayer? Deklarer resten av datastrukturen og tenk initialisering - når/ hvor leser du data fra fil, og hvordan representeres en tom hybel i datastrukturen din?
4. Deklarer metoder, i første omgang metodehodene (returverditype, navn, parametre).
5. Skriv/ skisser main()-metoden
6. Skriv/ skisser metodene
7. Gjenta punktene ovenfor til programmet er ferdig – du vil sikkert finne lurere løsninger på tidligere punkter mens du jobber

For hvert punkt er det lurt å kompilere, det er også fornuftig å skrive metodene i en slik rekkefølge at du får testet underveis at programmet kan kjøres - og utretter det du forventer. Leser du fra fil får du for eksempel fylt datastrukturen din med data, mens kommandoen Skriv oversikt viser deg hva som ligger i datastrukturen. I begynnelsen kan du også skrive ut løpende på skjermen det du leser inn fra fil, slik at du kan kontrollere at datafilen blir lest riktig. Dette fjernes når du er ferdig med testing.

Her følger noen pekere til stoff som er spesielt relevant/ nyttig for denne oppgaven:

1. **Klasser og objekter**, f.eks. kapittel 8 i læreboka eller Marit Nybakkens notat "objekter.pdf"; eller løse ukeoppgaver 6 og 7.

2. **Konstruktør:** Du kan lese mer om konstruktører i avsnitt 8.7 og 8.11 i læreboken, eller i forelesningsnotatene fra uke 7, eller i Marit Nybakkens notat «konstruktører.pdf».

3. **null-pekere:** Se læreboken kap 8.9 og *Ukeoppgaver 7*.

4. **Hybelnavn:** Disse består av et tall og en bokstav (uten noe skilletegn imellom), men slik det er mest naturlig å representere hyblene i datastrukturen vår (array) trenger vi både etasje og bokstav som heltall fra 0 og oppover. Dermed trenger vi å konvertere det vi leser og skriver **mellom char og int**, se typekonvertering i 2.4.2.

   Metoden **nextChar ()** i easyIO leser forbi skilletegn (blanke og ”usynlige” tegn som linjeskift etc) og returnerer neste karakter som ikke er et skilletegn. Nedenfor et eksempel på innlesing ved hjelp av **nextChar()** og konvertering **fra char til int** som tar hensyn til at arrayer har første indeks 0:

```
skjerm.out("Oppgi hybelnavn: ");

int etg = (int) (tast.nextChar() - '1'); // '1' gir [0], '2' gir [1], osv.

char bokstav = tast.nextChar(); // Les rombokstaven

int rom = (int) (bokstav - 'A'); // 'A' gir [0], 'B' gir [1], osv.
```

Etter disse setningene vil variabelen rom inneholde verdien 0 hvis brukeren tastet inn en A-hybel (f.eks. 3A), 1 hvis det var en B-hybel, osv.; og etasjenummer vil havne i variabelen etg (fratrukket 1).

Omvendt kan man konvertere **fra int til char** slik:

```
char bokstav = (char) ('A' + intVariabel);
```

5. **Filhåndtering.** Husk å lukke filen når du er ferdig med å skrive til den med fil.close(). Du kan forutsette at datafilen finnes i programmet ditt (du trenger ikke teste på det ved åpning) – sørg derfor for å kopiere datafilen hybeldata.txt til riktig område før du starter programmet.

   For å skrive på slutten av en fil uten å ødelegge det som ligger der fra før bruker du *append*-modus i EasyIO ved å angi en ekstra-parameter når du åpner filen: Out fil = new Out ("hole.txt", true);. Dersom denne filen ikke eksisterer fra før vil den bli opprettet.