

# Obligatorisk oppgave nr. 3 (av 4) i INF1000, våren 2006

## Advarsel

Etter forelesningen 6. mars har vi gjennomgått alt stoffet som trengs for å løse oppgaven. Du kan imidlertid godt starte arbeidet allerede nå, da du allerede har nok grunnlag til å håndtere mesteparten av kodingen. Siden dette er en arbeidskrevende oppgave, anbefaler vi at du starter på den så tidlig som mulig.

## Leveringsfrist

Oppgaven må leveres senest **fredag 17. mars kl 16.00**. Viktig: Se siste side av oppgaveteksten for detaljerte leveringskrav.

## Formål

Formålet med denne oppgaven er å gi trening i bruk av klasser, metoder, forgreninger, løkker, arrayer, kommunikasjon med bruker via terminal, og enkel filhåndtering.

## Oppgave

Utenfor kysten av Ruritania er det et område med store oljeforekomster under havbunnen, og myndighetene i landet har bestemt seg for å selge rettighetene til å utvinne olje til oljeselskaper. Det aktuelle havområdet er rektangulært og er delt opp i et rutenett med 7 x 12 ruter (se figur under), hvor radene er nummerert fra 0 til 6 og kolonnene er nummerert fra 0 til 11. Hver rute kalles et utvinningsfelt (eller bare felt). Hvert felt har et entydig navn på formen  $\text{Blokk}(\text{radnr}, \text{kolnr})$  hvor  $\text{radnr}$  er et heltall mellom 0 og 6 og  $\text{kolnr}$  er et heltall mellom 0 og 11. For eksempel angir  $\text{Blokk}(0,0)$ ,  $\text{Blokk}(0,1)$ ,  $\text{Blokk}(0,2)$ , ...,  $\text{Blokk}(0,11)$  feltene i øverste rad (fra venstre mot høyre) i figuren under.

Feltene (eller mer presist: utvinningsrettighetene til dem) legges ut for salg enkeltvis. Det er ingen grenser for hvor mange felter et oljeselskap kan eie, men et felt kan ikke ha mer enn ett oljeselskap som eier (det vil si at hvert felt er enten ikke solgt eller det eies av ett oljeselskap alene).

	0	1	2	3	4	5	6	7	8	9	10	11
0												
1												
2												
3												
4												
5												
6												

Myndighetene i Ruritania trenger nå et datasystem som kan holde rede på hvilke felter som er solgt og hvilke eiere disse har. Systemet skal også holde rede på hvor mye olje (målt i antall fat) som er utvunnet i

hvert felt. Opplysningene om hvert felt skal representeres ved et objekt av typen `FeltData`. Denne klassen skal ha følgende grensesnitt:

```
class FeltData {
    String eier;
    int utvunnet;

    FeltData () { ...} // Opprett felt uten eier og uten utvinning av olje

    void settEier (String e) {... } // Legg inn nytt eiernavn
    String hentEier () {...}      // Les av eiernavn

    void adderUtvinnig (int antfat) { ...} // Øk total utvinning med gitt antall fat
    int hentUtvinnig () { ... }          // Les av total utvinning i antall fat
}
```

Hvis du finner det hensiktsmessig, kan du utvide klassen med flere metoder og variable.

Brukeren av systemet vil være en funksjonær som jobber for myndighetene i Ruritania. Når et oljeselskap ønsker å kjøpe et felt, ringer de denne personen som så bruker ditt program for å finne et ledig felt (hvis det er noen) og registrere salget av feltet. Funksjonæren skal også kunne få ut fra programmet en oversikt over antall solgte felter og en del annen statistikk (se under).

Programmet skal være kommandostyrt. Det vil si at programmet skal kunne ta imot en kommando fra brukeren, utføre kommandoen, ta imot ny kommando, osv, helt til brukeren ønsker å avslutte. Mer konkret skal programmet oppføre seg slik sett fra brukerens side:

1. Det skriver ut på skjermen hvilke kommandoer brukeren kan gi.
2. Deretter ber programmet om en kommando fra brukeren og leser denne inn.
3. Programmet utfører den valgte kommandoen.

Programmet skal gjenta de tre trinnene ovenfor helt til brukeren gir kommando for å avslutte kjøringen (se nedenfor). For at programmet skal bli oversiktlig og velstrukturert, skal du legge vekt på en fornuftig inndeling av programkoden i form av metoder. Du skal derfor bruke minst en metode for hver kommando. Brukeren skal kunne gi følgende kommandoer:

- **Kjøp et felt.** Denne kommandoen vil funksjonæren gi hvis et oljeselskap ringer og sier at de ønsker å kjøpe et felt. Programmet skal da først sjekke om det er noen ledige felter (det vil si felter som ikke allerede er solgt). Hvis det ikke er noen ledige felter, må programmet gi brukeren beskjed om dette. Hvis det finnes ledige felter, skal programmet spørre om navnet på feltet som ønskes kjøpt og navnet på det oljeselskapet som ønsker å kjøpe feltet og registrere at dette selskapet nå eier det aktuelle feltet. Her antar vi at de ulike oljeselskapene har forskjellige navn, slik at navnet entydig identifiserer et oljeselskap). Programmet skal også skrive ut på skjermen at salget gikk i orden, for eksempel med en utskrift som dette: "Blokk(3,5) selges til Petrol A/S". Hvis feltet allerede var solgt, må programmet i stedet skrive ut en feilmelding.

- **Annuller kjøp av et felt.** Denne kommandoen vil funksjonæren gi hvis et oljeselskap ringer og sier at de ønsker å annullere kjøpet av et felt. Programmet må da spørre brukeren om navnet på oljeselskapet og navnet på feltet som oljeselskapet ikke lenger ønsker å eie. Deretter skal programmet sjekke at dette feltet faktisk er solgt til dette oljeselskapet, og i så fall registrere at feltet ikke lenger har noen eier (og følgelig er ledig for

salg). Mengden olje som er utvunnet i feltet skal ikke settes til null igjen. Brukeren skal få bekreftelse på at salget er annullert. Hvis feltet derimot ikke var solgt til dette oljeselskapet, må brukeren få beskjed om dette.

- **Lag oversiktskart.** Programmet skal da skrive ut på skjermen et kart hvor hvert felt er markert med en "O" hvis det er ledig og "X" hvis det er solgt.
- **Lag liste over oljeselskaper.** Programmet skal da skrive ut på skjermen en liste med navnene på alle oljeselskapene som eier felter. Hvert oljeselskap skal bare forekomme en gang i denne listen, og under navnet på hvert oljeselskap skal det være en liste over alle feltene selskapet eier.
- **Finn oljeselskapet med flest felt.** Programmet skal da finne ut hvilket oljeselskap som eier rettigheter på flest felt. Navnet på selskapet skal skrives ut på skjermen sammen med antall tilnyttede oljefelt og hvor mange fat olje som er utvunnet til sammen på disse feltene. Hvis det ikke er entydig hvilket selskap som driver flest felt (flere selskap har like mange felter), skal programmet skrive ut informasjon for kun ett av disse selskapene (vilkårlig hvilket).
- **Oppdater oljeutvinning.** Annenhver måned må oljeselskapene ringe til funksjonæren og fortelle hvor mye olje de har utvunnet på hvert av feltene i løpet av de to siste månedene. Da bruker funksjonæren denne kommandoen for å registrere informasjonen. Programmet ber da om navnet på feltet og hvor mye olje (målt i fat) selskapet har utvunnet der de siste to månedene, og oppdaterer registeret.
- **Finn gjennomsnittlig oljeutvinning.** Programmet skal da regne ut hvor mye olje som er utvunnet i gjennomsnitt blant de feltene som er solgt, og skrive ut resultatet på skjermen. Programmet skal altså *ikke* ta med i gjennomsnittet de feltene som ikke har noen eier.
- **Avslutt.** Programmet skal da avslutte. Programmet skal ta vare på (skrive til fil) de dataene som er registrert av funksjonæren, se nærmere beskrivelse etter kjøre-eksempelet.

Nedenfor ser du et eksempel på starten av en kjøring av programmet (bruker-input er markert med understreking). Utskriften er bare ment å illustrere hvordan kommunikasjonen kan foregå. Dersom du ønsker å presentere menyen annerledes, osv, så kan du gjøre det.

```
>java Oblig3
```

```
*** VELKOMMEN TIL RURITANIAS OLJEFELTOVERSIKT ***
```

```
Du har følgende valgmuligheter:
```

- 1) Kjøp et felt
- 2) Annuller kjøp av et felt
- 3) Lag oversiktskart
- 4) Lag liste over oljeselskaper
- 5) Finn oljeselskapet med flest felt
- 6) Oppdater oljeutvinning
- 7) Finn gjennomsnittlig oljeutvinning
- 8) Avslutt

```
Velg kommando: 1
```

```
** Kjøp et felt **
```

```
Oppgi feltet som ønskes kjøpt: Blokk(3,6)
```

```
Oppgi oljeselskapets navn: Oljesøl A/S
```

```
Blokk(3,6) selges til Oljesøl A/S.
```

Du har følgende valgmuligheter:

- 1) Kjøp et felt
- 2) Annuller kjøp av et felt
- 3) Lag oversiktskart
- 4) Lag liste over oljeselskaper
- 5) Finn oljeselskapet med flest felt
- 6) Oppdater oljeutvinning
- 7) Finn gjennomsnittlig oljeutvinning
- 8) Avslutt

Velg kommando: 5

.... osv ....

For at opplysningene som er registrert i systemet ikke skal gå tapt når programmet stoppes, må programmet kunne lese og skrive data fra og til en fil. Ved oppstart av programmet skal det lese inn data som finnes i filen `ruritania-olje.dat`. Når programmet avsluttes (kommando nr. 8 i listen ovenfor), skal denne filen skrives på nytt basert på datasettet som nå ligger i den to-dimensjonale arrayen. En eksempel-fil kan du finne samme sted som du lastet ned denne oppgaveteksten. Filen (både ved lesing og skriving) skal ha følgende format:

```
<antall felt som er i bruk i arrayen>

<rad-indeks i array> <kolonne-indeks i array>
<eiernavn for første registrerte felt>
<antall fat utvunnet for første registrerte felt>

...

<rad-indeks i array> <kolonne-indeks i array>
<eiernavn for siste registrerte felt>
<antall fat utvunnet for siste registrerte felt>
```

## Tips

Det finnes et vedlegg til denne oppgaveteksten som foreslår mulige angrepsvinkler for de ulike punktene i oppgaven. Dette vedlegget kan lastes ned fra kursets hjemmeside, <http://www.ifi.uio.no/inf1000/v06>.

## Leveringskrav

Du plikter å ha lest og forstått følgende krav til innleverte oppgaver ved institutt for informatikk:

<http://www.ifi.uio.no/studinf/skjemaer/erklaring.pdf>

Oppgaven skal leveres elektronisk. For godkjenning kreves det at koden er kommentert og pent formatert med passende innrykk.

Første linje i programfilen (.java-filen) skal se slik ut:

```
// dittBrukernavn g-gruppeNummer o-3 k-inf1000/v05
```

hvor *dittBrukernavn* skal erstattes av ditt eget brukernavn (det navnet du oppgir når du logger deg inn på UiO's anlegg), og *gruppeNummer* skal erstattes av nummeret på den øvingsgruppa du er tatt opp på (hvis du har byttet øvingsgruppe, skal du altså oppgi gruppenummeret til den opprinnelige øvingsgruppa). Eksempel: hvis ditt brukernavn er pelle og du skal levere oblig3 på gruppe 4, så skal første linje i programfilen se slik ut:

```
// pelle g-4 o-3 k-inf1000/v06
```

Send programfilen (.java-filen) og resultatet av en testkjøring (se nedenfor) med e-post til gruppelæreren din (inf1000-*gruppeNummer*@ifi.uio.no). Resultatet av testkjøringen skal vise utskriften på skjermen når du starter opp programmet ditt og tester ut de forskjellige delene som du er spurt om å lage i oppgaven. Kjører du Unix, kan du bruke et program som heter *photo* for å lagre på fil alt som skrives ut på skjermen under en testkjøring. Du gir da følgende kommando i xterm-vinduet rett før du starter java-programmet ditt:

```
photo testOblig3.txt
```

Etter at du har startet programmet ditt, testet det ut og fått det til å avslutte, gir du kommandoen *exit* (eller trykker Control-D) for å avslutte photo-programmet. Nå ligger all utskriften fra testen på filen *testOblig3.txt* (du kan kalle filen noe annet hvis du ønsker).

Hvis du har spørsmål vedrørende leveringsmåte eller annet, så kontakt gruppelæreren din i god tid før innleveringsfristen. Det er ditt ansvar at oppgaven kommer frem til øvingslæreren på riktig måte innen leveringsfristen.

**Lykke til!**