

# Oblig2 - obligatorisk oppgave nr. 2 (av 4) i INF1000 – h2007

### Leveringsfrist

Oppgaven må løses individuelt og leveres senest fredag 28. september kl 16.00 via Joly. Viktig: les slutten av oppgaven for detaljerte leveringskrav.

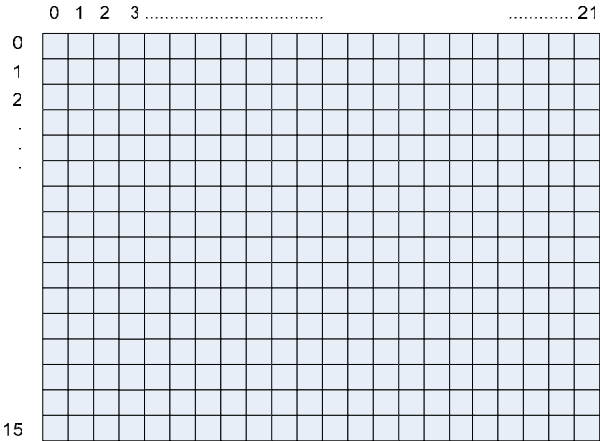
### Formål

Formålet med denne oppgaven er å gi trening i bruk av forgreninger, løkker, arrayer og kommunikasjon med bruker via terminal – eventuelt også bruk av metoder.

### Oppgave

Utenfor kysten av Ruritania er det et område med store oljeforekomster under havbunnen, og myndighetene i Ruritania 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 16 x 22 ruter (se figur under), hvor radene er nummerert fra 0 til 15 og kolonnene er nummerert fra 0 til 21. Hver rute kalles et utvinningsfelt (eller bare felt). Hvert felt har et entydig navn på formen  $F(radnr, kolnr)$  hvor  $radnr$  er et heltall mellom 0 og 14 og  $kolnr$  er et heltall mellom 0 og 21. For eksempel angir  $F(0,0)$ ,  $F(0,1)$ ,  $F(0,2)$ , ...,  $F(0,21)$  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 (dvs hvert felt er enten ikke solgt eller det eies av ett oljeselskap).



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. Brukeren av systemet skal være en funksjonær som jobber for myndighetene i Ruritania. Når et oljeselskap ønsker å kjøpe et felt, ringer de funksjonæren 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 andre ting (se under).

Programmet skal være kommandostyrt, dvs. det 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:

- Det skriver ut på skjerm hvilke kommandoer brukeren kan gi.
- Deretter ber programmet om og leser inn en kommando fra brukeren.
- Programmet utfører den valgte kommandoen.

Programmet skal gjenta de tre trinnene ovenfor helt til brukeren gir kommando om å avslutte (se nedenfor). 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 (dvs. 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 (vi gjør den rimelige antagelsen at de ulike oljeselskapene har forskjellig navn, slik at navnet entydig identifiserer et oljeselskap) og registrere at dette oljeselskapet nå eier det aktuelle feltet. Programmet skal også skrive ut på skjermen at salget gikk i orden, f.eks. med en utskrift som dette: "Feltet F(13,15) selges til Petrol A/S". Hvis feltet allerede var solgt, må programmet i stedet gi 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 (se eksempel lenger ned i oppgaven) hvor hvert felt er markert med en "." 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 angis hvor mange (men **ikke** hvilke) felter selskapet eier.

• **Oppdater oljeutvinning.** Hver tredje måned må oljeselskapene ringe til funksjonæren og fortelle hvor mye olje de har utvunnet på hvert av feltene i løpet av de tre 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 tre måneder, og oppdaterer registeret.

• **Finn maksimal oljeutvinning.** Programmet skal da finne ut hvilket felt som har utvunnet mest olje, og skrive ut på skjermen navnet på dette feltet, navnet på eieren og hvor mye olje som er utvunnet der. Hvis flere felter tilfredsstillt kravet, holder det at programmet skriver ut informasjon for ett av disse feltene (vilkårlig hvilket).

• **Finn sum oljeutvinning.** Programmet skal da regne ut hvor mye olje som er utvunnet i sum i alle de feltene som er solgt, og skrive ut resultatet på skjermen. Husk å få med alle feltene, også de som er levert tilbake, men hvor det har vært utvinning.

• **Avslutt.** Programmet skal da avslutte. Det er ikke meningen at programmet skal ta vare på (skrive til fil) de dataene som er registrert av funksjonæren.

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 Oblig2
```

```
*** 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) Oppdater oljeutvinning
- 6) Finn maksimal oljeutvinning
- 7) Finn sum oljeutvinning
- 8) Avslutt

```
Velg kommando: 1
```

```
** Kjøp et felt **
```

```
Oppgi feltet som ønskes kjøpt: F(3,6)  
Oppgi oljeselskapets navn: Oljesøl A/S  
Feltet F(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) Oppdater oljeutvinning
- 6) Finn maksimal oljeutvinning
- 7) Finn sum oljeutvinning
- 8) Avslutt

```
Velg kommando: 5
```

```
.... osv ....
```

### Tips og forutsetninger som kan gjøres

For hvert felt er det to opplysninger som skal tas vare på: (1) navnet på oljeselskapet som eier det; og (2) hvor mye olje som er utvunnet i feltet. Du kan bruke to to-dimensjonale arrayer til å lagre disse opplysningene:

```
String[][] eier = new String[16][22];  
double[][] utvunnet = new double[16][22];
```

Dermed vil f.eks. `eier[3][5]` være navnet på oljeselskapet som eier feltet F(3,5), og `utvunnet[3][5]` vil være antall fat olje som er utvunnet på dette feltet (sistnevnte array kunne også vært deklarerert som en int-array - her står du fritt til å velge). Det finnes andre måter å løse oppgaven på (vi kunne klart oss med en enkelt to-dimensjonal array med pekere til objekter av en klasse som vi selv lager - men denne teknikken gjennomgås ikke før senere i kurset og skal ikke anvendes i denne oppgaven).

Her er noen tips og kommentarer til noen av kommandoene. Hver av disse punktene løses lettest hvis du for hvert punkt lager en metode som kalles hvis brukeren ønsker den kommandoen utført, men siden metoder er relativt nytt stoff, aksepteres også løsninger uten slike metoder :

### •Kjøp et felt.

Programmet må ha en måte å finne ut om et felt er solgt eller ikke. En enkel måte å finne ut om f.eks. F(3,6) er solgt er å se om `eier[3][6]` er satt til et navn eller ikke. Rett etter at 2D-arrayen `eier` er deklarerert er alle verdiene i arrayen lik den spesielle verdien `null`, og du kan derfor sjekke om F(3,6) er ledig ved å teste om `eier[3][6] == null`.

Når programmet skal lese inn navnet på feltet som skal selges, må du klare å trekke ut radnummeret og kolonnennummeret fra feltnavnet. Hvis f.eks. brukeren skriver "F(3,6)" så må du altså klare å "få tak i" tallene 3 og 6 siden du trenger disse når du skal sette inn navnet på oljeselskapet i `eier`-arrayen. Dette kan gjøres på flere måter; en av dem er å bruke setningene

```
int radnr = tast.inInt("F(, )");
int kolnr = tast.inInt("F(, )");
```

Her er det en variant av `inInt()` som brukes. Det som står i anførselstegn er de tegnene som skal betraktes som skilletegn - altså de tegnene som `inInt` skal se på som "blanke tegn". I tillegg får vi alltid med linjeskift og vognretur (men ikke 'blank') som skilletegn når vi angir egne skilletegn som ovenfor.

### •Annuler kjøp av et felt.

Her må du huske å sette `eier[radnr][kolnr]` til `null`, hvor `radnr` og `kolnr` er radnummeret og kolonnennummeret til det feltet som ikke lenger skal ha noen eier.

### •Lag oversiktskart.

Det oversiktskartet som skal lages skal se omtrent slik ut (men hvor plasseringen av "x" og "." naturligvis kan variere og skal vise hvilke felter som er henholdsvis solgt og ledig).

```

  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
0 x  x  x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1 x  x  x  .  .  .  .  .  .  .  .  .  .  x  x  .  .  .  .  .  .  .
2 .  x  x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  x  .
3 .  x  x  x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  x  x  .
4 .  .  x  .  .  .  .  .  .  .  x  .  .  .  .  .  .  .  .  .  .  .  .
5 .  .  x  .  .  .  .  .  .  .  x  x  .  .  .  .  .  .  .  .  .  .  .
6 .  .  x  .  .  .  .  .  .  .  x  .  .  x  .  .  .  .  .  .  .  .  .
7 .  .  x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
8 .  .  x  x  .  .  .  .  .  .  .  .  .  .  x  .  .  .  .  .  .  .  .  .
9 .  .  x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
10 .  .  x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
11 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
12 .  .  x  .  .  .  .  .  .  .  x  .  .  .  .  .  .  .  .  .  .  .  .
13 .  .  x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
14 .  .  x  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
15 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

- **Lag liste over oljeselskaper med antall felter.**

Her lønner det seg å først lage en String-array med navnene på oljeselskapene som eier felter, og hvor hvert selskap bare forekommer en enkelt gang i arrayen. Det er to måter å gjøre dette på. Den ene måten er å lage en slik array når programmet starter opp, og sørge for å legge inn navn på nye oljeselskap i arrayen når kommandoen "Selg et felt" utføres (du må da også huske å ta ut et navn dersom et oljeselskap annullerer et kjøp og ikke lenger eier noen felter). Den andre måten er å lage en slik array hver gang kommandoen "Lag liste over oljeselskaper" utføres. Da må du løpe gjennom hele 2D-arrayen **eier**, og hver gang du ser et navn som ikke allerede ligger i arrayen over oljeselskapnavn så må du legge det inn i sistnevnte array som eier av ett felt. Hvis du senere treffer på et oljeselskap som du allerede har navnet på, må du så telle opp antall oljefelter det selskapet eier.

- **Oppdater oljeutvinning.**

Husk at du i denne oppgaven skal addere den innleste verdien til den verdien som allerede er registrert i systemet (siden verdien i registeret skal være det totale antall fat olje som er utvinnet i et felt).

- **Finn maksimal oljeutvinning.**

Denne oppgaven løser du greiest ved å løpe gjennom hele 2D-arrayen **utvunnet** og hele tiden holde rede på den største verdien du har sett hittil (og hvor du så den). Husk at du må hoppe over de feltene som ikke har noen eier.

- **Finn sum oljeutvinning.**

Her kan du bare løpe gjennom hele 2D-arrayen utvunnet og summere verdiene.

Du kan gjøre egne forutsetninger etter behov så lenge disse ikke er i strid med oppgaven, men sørg for å kommentere disse i programmet. Et tips: sørg først for å få kommandoløkken til å fungere, dvs sørg for at programmet klarer å skrive ut lovlige kommandoer og lese inn en kommando på korrekt måte. Deretter kan du gå videre og programmere de enkelte kommandoene (i vilkårlig rekkefølge).

Husk å compilere og teste programmet ditt regelmessig når du lager det!

## **Leveringskrav**

Her student skal levere hver sin egenprogrammerte løsning, og 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 (java-filen) skal leveres elektronisk i Jolysystemet

<http://obelix.ifi.uio.no:8080/wizard.html>

Da må du legge en kommentar som første linje i .javafilen din som sier hvem du er, login-navnet, gruppe, obliqnummer, kurskode og semester som flg:

```
//<brukernavn> g-<gruppernummer> o-<obliqnummer> k-<kurskode>/<semester>.
```

eks: for Arne Maus, (arnem) på gruppe 12: // arnem g-12 o-2 k-inf1000/h07 ).

Se også kursets hjemmesider for oppdateringer og beskjeder.