

# Oblig2 - obligatorisk oppgave nr. 2 (av 4) i INF1000 – høst 2011

---

## Leveringsfrist

Oppgaven må løses individuelt og leveres senest fredag 30. september kl 16.00 via Joly.  
Viktig: les slutten av oppgaven for detaljerte leveringskrav.

## Formål

Å gi studenten erfaring i å designe og skrive programmer med objektorientert struktur, samt trening i bruk av metoder og andre Java-elementer fra pensum.

## 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 8 x 20 ruter (se figur under), hvor radene er nummerert fra 0 til 7 og kolonnene er nummerert fra 0 til 19.

Hver rute kalles et utvinningsfelt (eller bare felt). Hvert felt har et entydig navn på formen *radnr-kolnr* hvor *radnr* er et heltall mellom 0 og 7 og *kolnr* er et heltall mellom 0 og 19. For eksempel angir 0-0, 0-1, 0-2, ..., 0-19 feltene i øverste rad (fra venstre mot høyre) i figuren under.

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

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 trenger også andre funksjoner i programmet (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 3-15 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 nedenfor) hvor hvert felt er markert med en "." hvis det er ledig og "x" hvis det er solgt. NB: Det er ikke et krav at markeringene i samme kolonne kommer rett under hverandre, så lenge alle kolonnene er med i hver rad som skrives ut.

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

- **Oppdater oljeutvinning.** Hvert år må oljeselskapene ringe til funksjonæren og fortelle hvor mye olje de har utvunnet på hvert av feltene i løpet av det siste året. 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 siste år, og oppdaterer registeret.
- **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) Oppdater oljeutvinning
- 5) Finn sum oljeutvinning
- 6) Avslutt

```
Velg kommando: 1
** Kjøp et felt **
Oppgi feltet som ønskes kjøpt: 3-6
Oppgi oljeselskapets navn: Oljesøl A/S
Feltet 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) Oppdater oljeutvinning
- 5) Finn sum oljeutvinning
- 6) Avslutt

Velg kommando: 5

.... OSV ....

## Krav til løsning

- Løsningen skal inneholde en metode for hver av de tillatte kommandoene, samt en for innlesing av kommando, og benytte to klasser (en med main-metoden og en med kommandoløkken, data for foljefeltene og alle metodene som gjør de ulike kommandoene).
- Datastrukturen som representerer oljefeltet, og alle metoder som bearbeider denne, skal ligge i en egen klasse

## Relevant pensumstoff

Følgende anbefales sterkt å repetere før du starter planlegging/ design av programmet ditt:

- Design av program med metoder, kap 7.3

Andre deler av pensum du antakelig vil benytte er (ikke komplett!):

- Metoder, kap 7.1-7.7
- Todimensjonale arrayer, kap. 5.7
- Terminal I/O, kap 3.3-3.5
- String, kap. 6.4.1

## Krav til innlevering

Oblig2 skal leveres (java-filen) innen fredag 30. sept. kl. 1600 til Joly.

## Tips om programstrukturen

Nedenfor følger en skisse av hvordan du kan starte på Oblig2, med to klasser (Oblig2 og Olje). Dette er bare et tips, du kan gjerne gjøre det på 'din' måte, men denne obligen **skal** løses med to klasser og én metode for hver mulige ordre.

```
import easyIO.*;
class Oblig2 {
```

```

public static void main(String[] args) {
    Olje ol = new Olje();
    ol.ordreløkke(); // Kjører metoden ordreløkke i klassen Olje
    System.out.println("-- Programmet avslutter --");
}
}

class Olje {
    // Klargjøring for innlesing/utskrift, gjelder for hele klassen:
    In tast = new In();
    Out skjerm = new Out();
    // Her kan du deklareere arrayene eier[][] og utvunnet[][]:
    // Metoden "ordreløkke()":
    // Denne er den første metoden i klassen Olje som blir utført.
    void ordreløkke() {
        int ordre = 0;

        // 6 betyr avslutt
        while (ordre != 6) {
            // Legg her setninger som skriver ut menyen, f.eks. linjer med
            // println eller skjerm.outln(), eller kall på en egen metode.
            System.out.println(" 1) Liste over felt med eier");

            // Her kan du skrive ut en ledetekst, f.eks. "Velg ordre: ",
            // og så leser du input som bruker taster inn.
            ordre = ...; // (Erstatt ... med kode som leser fra tastatur.)

            switch (ordre) {
                case 1: kjøpEtFelt(); break;
                case 2: annullerKjøp(); break;
                case 3: lagOversiktskart(); break;
                // ... fyll inn case-er for de to andre ordrene her.
                default: break;
            }
        }
    }
}

```

```
// De andre metodene: (en for hver ordre)
void kjøpEtFelt() {
    // Programmér kjøp av felt her:
    // - Be bruker taste inn rad-kol., og oljeselskap.
    // - Test om arrayen eier[][] allerede har et oljeselskap i den
    //   angitte rad-kol.: (a) Hvis ikke => registrér kjøpet, osv...
}
void annulerKjøp() {
}
void lagOversiktskart() {
}
// ...Osv.(resten av metodene - én for hver av de gjenstående ordrene)
```