

Vedlegg til obligatorisk oppgave 3 i INF1000, våren 2006

Dette dokumentet inneholder råd om hvordan obligatorisk oppgave 3 kan løses. Oppgaven kan løses på andre måter, men rådene nedenfor vil nok være en fornuftig angrepsvinkel for de fleste.

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. Disse opplysningene skal lagres i et objekt av typen `FeltData` for hvert felt. Du kan bruke en to-dimensjonal array til å lagre disse opplysningene:

```
FeltData[][] blokk = new FeltData[7][12];
```

Dermed vil f.eks. `blokk[3][5]` være et `FeltData`-objekt som inneholder navnet på oljeselskapet som eier feltet `Blokk(3,5)`, samt antall fat olje som er utvunnet på dette feltet.

Her er noen tips og kommentarer til noen av kommandoene:

- **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. `Blokk(3,6)` er solgt er å se om `blokk[3][6]` inneholder et navn eller ikke. Rett etter at 2D-arrayen `blokk` er deklartert, er tekststrengen `eier` for alle objektene i arrayen satt den spesielle verdien `null`, og du kan derfor sjekke om `Blokk(3,6)` er ledig ved å teste om `blokk[3][6].hentEier() == 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 "`Blokk(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 `blokk`-arrayen.

Dette kan gjøres på flere måter; en av dem er å bruke setningene

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

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".

- **Annuler kjøp av et felt.**

Her må du huske å sette `eier` i `blokk[radnr][kolnr]` til `null`, hvor `radnr` og `kolnr` er radnummeret og kolonnennummeret til det feltet som ikke lenger skal ha noen eier. Ved skrivning til fil vil en tekststreng som har verdien `null` resultere i utskrift av teksten "null". Hvis denne opplysningen skal leses inn igjen senere, vil verdien i variabelen for innlesing (her `eier`) bli teksten "null". Dette er ikke det samme som at variabelen settes til `null`. Ved innlesing av tekststrengen for eier må man derfor teste om teksten "null" ble lest, og i så fall eksplisitt sette `eier` til verdien `null` for det aktuelle feltet.

- **Lag oversiktskart.**

Det oversiktskartet som skal lages skal se omtrent slik ut (men hvor plasseringen av "X" og "O" 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
```

```

0  X  X  X  O  O  O  O  O  O  O  O  O
1  O  O  O  O  O  X  X  X  X  X  X  X
2  O  X  X  O  X  X  X  O  O  O  O  O
3  X  X  X  O  O  O  O  O  O  O  O  O
4  O  O  O  O  O  X  X  X  X  X  X  X
5  X  X  X  O  O  O  O  X  X  O  O  O
6  X  O  O  O  O  O  X  X  X  O  O  O

```

- **Lag liste over oljeselskaper.**

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 **blokk**, og hver gang du ser et navn som ikke allerede ligger i arrayen over oljeselskapnavn så må du legge det inn i sistnevnte array.

- **Finn oljeselskapet med flest felt.**

Denne oppgaven løser du greiest ved samme framgangsmåte som i punktet ovenfor (liste over selskaper), hvor du også summerer opp antall felt for hvert oljeselskap. Til denne oppsummeringen trenger du en int-array `antall_felt` som matcher String-arrayen ovenfor både mht. lengde og rekkefølge. Oljeselskapet gitt ved navnet i posisjon `k` i String-arrayen skal da ha rettigheter på totalt `antall_felt[k]` oljefelt.

- **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 gjennomsnittlig oljeutvinning.**

Her må du også løpe gjennom hele 2D-arrayen utvinnet, men for hvert felt må du i tillegg sjekke om det har noen eier, siden gjennomsnittet bare skal beregnes på bakgrunn av de feltene som er solgt.

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, det vil si 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!