

Oblig 4 (av 4) INF1000, høsten 2012
Værdata, leveres innen 9. nov. kl. 23.59

Formål

Formålet med denne oppgaven er å gi trening i hele pensum og i å lage et større program. Løsningen du lager skal være Objekt-Orientert. Det innebærer blant annet at datastrukturen (den delen av programmet som skal "holde på dataene") skal gjøre bruk av objekter av klasser som du selv definerer. *Oppgaven skal besvares individuelt – enhver kopi av hele eller deler av andres løsninger vil bli behandlet som fusk!* Du er forpliktet til å sette deg inn i [krav og regler](#) for obligatoriske oppgaver.

Husk at din besvarelse skal leveres som én .java-fil til Joly (alle klassene i samme fila). For mer informasjon om hvordan du kan gå fram for å løse oppgaven, se avsnittet *Tips & krav til løsningen*.

Oppgave

Meteorologisk institutt har en rekke værstasjoner rundt i Norge. Ved hver stasjon gjøres det daglig en rekke værmålinger en eller flere ganger, og resultatene samles på store datafiler for senere analyser. Hensikten med denne oppgaven er å lage et program som kan lese slike meteorologiske data fra fil, legge dataene inn i en fornuftig datastruktur, gjøre noen relativt enkle beregninger og skrive ut resultatene på skjerm. Programmet skal være kommandostyrt.

Du skal i denne oppgaven bruke data som du finner på to filer: "[klima_data2012.txt](#)" og "[stasjoner_norge.txt](#)", som du finner og laster ned fra hjemmesida til INF1000.

Ser vi først på fila: "`stasjoner_norge.txt`", ser vi at den i linje 2 til 40 har opplysninger om 39 værstasjoner i Norge, og av overskrifta i linje 1 ser vi at dette er: Et entydig stasjonsnummer, stasjonens navn, stasjonens høyde over havet, kommune, fylke og region for denne stasjonen.

Når du leser inn data om en stasjon, lager du et objekt av klassen Stasjon, og så legger du det inn i en HashMap av stasjoner med stasjonsNummer som nøkkel (se hint c). Du skal også lage en array av HashMap-er med en HashMap for hver region (ØSTLANDET, AGDER, NORD-NORGE, VESTLANDET, TRØNDELAGE, NORGE) (se hint h). Det stasjon-objektet du har lest inn, legger du også inn i et av disse HashMap-ene, her med stasjonsnavnet som nøkkel. Kanskje vil du nytte en oppramstype (Enum) over region-navnene for lett å finne den riktige indeksen inn i denne HashMap-arrayen når du har region-navnet.

Alle data på disse 40 linjene skal du lese inn i programmet ditt (Hint, bruk: `readLine()` til å hoppe over første linje) og inn i datastrukturen din (se nedenfor).

Fila "`klima_data2012.txt`" inneholder observasjoner fra alle dagene i jan-sept 2012 for de 39 værstasjonene. **N.B.** Mangler noen av observasjonene er det markert med verdien -999. Programmet ditt må sørge for ikke å ta med denne -999 verdien når det sammenlignes verdier eller regnes ut gjennomsnitt. Den tiende linja betyr da at:

```
19710 1 1 7 -0.5 -999 10.0 160 2.6
```

PUBLISERT FREDAG 26.10 (diff fra foreløpig versjon: korrektur og tips). Rettelse 28.10: endret tips h)

på stasjon 19710 (ASKER), var det den 01.01.2012 kl.7 en lufttemperatur -0.5 grader Celsius, data om nedbør siste 24timer mangler, nedbøren siste 12 timer var 10,0 mm; og det blåser en vind fra retning 160 grader med en hastighet på 2,6 m per sekund. Merk deretter at det er fra fire til én linje med observasjoner for hver dag; er det fire er de for kl. 01, kl. 07, kl. 13 og kl. 19. På de 7 første linjene ligger det en beskrivelse av data på filen, linje 8 er blank og på linje 9 ligger det overskrifter for kolonnene i resten av fila.

Å regne ut nedbør

Merk at ”**klima_data2012.txt**” mangler mange nedbørsregistreringer. Vi ser at:

- a) Det er mange stasjoner (11 stk) som ikke har registrert nedbør i det hele tatt. Det kommer vel av at Meteorologisk institutt ikke samler inn nedbørsdata fra alle stasjonene. Det medfører at det er 3 hele fylker man ikke har nedbørsdata fra. Blir man bedt å beregne et gjennomsnitt man ikke har noen data fra får man skrive «Data mangler!» og returnere til kommandoløkken.
- b) For noen av stasjonene mangler data helt eller delvis for noen få dager. Det kommer av sykdom eller at måleapparatet var i stykker eller andre grunner gjorde at man ikke kunne foreta noen av observasjonene. Da må man se etter at man, når det f.eks. skal lages et gjennomsnittstall for en uke eller måned, bare tar med i summen de dagene man kan finne nedbørdata fra; og snittet blir da denne summen delt på det antall dager man har data fra.
- c) Beregning av nedbør for dager der det finnes minst én observasjon:
 - a. Finnes én observasjon av 24timers typen, brukes den
 - b. Finnes kun én observasjon av 12timers typen, brukes 2 ganger den verdien
 - c. Finnes to observasjoner av 12timers type, legges disse sammen og brukes
 - d. Alle andre tilfeller behandles som om alle data mangler for denne dagen.

Oppgave 1)

Hele denne oppgaven skal løses med klasser og objekter. Ut fra opplysningene ovenfor og ved å lese gjennom de spørsmål systemet skal kunne gi svar på i oppgavene 2-8, så skal du tegne et klassediagram i UML over systemet ditt. Gi navn på de forhold du vil ha i systemet. Skriv også antall på begge sidene av forholdene. Hvis du greier å få skannet inn denne tegningen, kan du levere den inn på e-post til gruppelæreren din - hvis ikke, skal du kunne levere den inn hvis hjelpelæreren ber om det.

Oppgave 2)

Du skal nå skrive de delene av programmet som leser de to filene, først linjene 2 til 40 fra ”**stasjoner_norge.txt**” og deretter linje 10 til 37221 fra ”**klima_data2012.txt**”. Du oppretter objekter av de klassene du har definert etter hvert som du leser filene. Alle klassene (med unntak av den som inneholder ’**main**’) skal ha en konstruktør du har skrevet selv og som initierer objektet med de data du leser inn om vedkommende objekt. Navnene på disse filene skal være parametere når du starter programmet (dvs. filnavnene er i **args[0]** og **args[1]** som parametere til **main** - se hint).

Oppgave 3)

Du skal nå utstyre programmet med en metode som skriver ut en meny for brukeren og som gir brukeren følgende valg, som du skal programmere i oppgavene 4-6:

- 1 - **finnAntallUversDager(int stasjonsNummer, int maaned)**
- 2 - **sammenlignVindVedKyststasjonerMotResten()**
- 3 - **sammenlignOstlandetMotNordNorge(int maaned)**

PUBLISERT FREDAG 26.10 (diff fra foreløpig versjon: korrektur og tips).
Rettelse 28.10: endret tips h)

- 4 - `finnKaldesteVerstasjonMaaned(int maaned)`
- 5 - `skrivUtVarmesteRegionForHverMaaned()` - (kan sløyfes)
- 6 - avslutt

Oppgave 4)

Du skal skrive metoden :

`finnAntallUversDager(int stasjonsNummer, int maaned)`

som går gjennom alle observasjonsdata for den brukervalgte værstasjonen og for den måned brukeren også har valgt, og summerer antall døgn hvor *summen av* nedbør i mm og vindhastighet i m/sek er større enn 10.7 (dvs. enten blåser det mye eller det regner mye – eller begge deler). Tallet 10.7 er valgt fordi hvis det ikke regner den dagen, så er det minst kuling i vindstyrken. Har du her ikke nedbørsdata, bruk verdien 0 for den dagen. Skriv ut stasjonsnavn, måned og antall dager med uvær du fant for denne stasjonen i den valgte måneden.

Oppgave 5)

Du skal skrive metoden :

`sammenlignVindVedKyststasjonerMotResten()`

som prøver å belyse en påstand at det blåser mer på værstasjoner som ligger ved kysten enn ved de andre stasjonene (som da ligger inne i landet). Vi skal definere at alle værstasjoner som ligger mindre enn 60 meter over havet, ligger ved kysten og resten ligger i innlandet. Gå gjennom alle data du har registrert og lag gjennomsnitt sekundmeter vind for kyststasjonene per dag (for alle de månedene du har data for) og skriv dette gjennomsnittet ut sammen med tilsvarende gjennomsnitt for ”innlandsstasjonene”. Skriv så ut en kommentar fra programmet som sier om påstandene om mer vind på kysten er riktig eller ikke.

Oppgave 6)

Du skal skrive metoden :

`sammenlignOstlandetMotNordnorge(int maaned)`

som i undersöker påstanden at det i gjennomsnitt regner mindre *og* samtidig er varmere på Østlandet enn i Nordnorge i den valgte måneden.

Du regner så ut og skriver ut på en egen fil ”**Resultat.txt**” for den valgte måneden gjennomsnittstemperatur, samt gjennomsnittlig nedbør for de to gruppene av værstasjoner (nordnorske og østlandske). Til slutt skriver programmert ditt ut en kommentar om påstanden ovenfor var riktig eller ikke.

Oppgave 7)

Skriv metoden

`finnKaldesteVerstasjonMaaned(int maaned)`

som går gjennom data for alle stasjonene for den gitte måned (data gitt av brukeren) og skriver ut navnet på den stasjonen hvor *gjennomsnittet av temperaturene* for hver dag den måneden er minst av samtlige stasjoner.

Oppgave 8) – ekstraoppgave (kan sløyfes)

Skriv metoden

`skrivUtVarmesteRegionForHverMaaned()`

som går gjennom alle regionene for hver av månedene i jan-sept 2012 og finner hvilken region som var varmest den måneden. På skjermen skal det i tillegg til en passende overskrift, skrives ut en linje

PUBLISERT FREDAG 26.10 (diff fra foreløpig versjon: korrektur og tips).
Rettelse 28.10: endret tips h)

for hver av de 9 månedene med månedens navn, regionens navn og den regionens snitt-temperatur den måneden.

Levering av følgende for Oblig-4 innen 9. nov kl. 23.59:

Du skal levere via Joly, som bare tar to filer. Hvis du *ikke* greier å lage en .zip fil så leverer du:

1. Programmet (.java filen med alle klassene)
2. Den html-filen som lages av javadoc av klassen Stasjon (husk å legge inn javadoc-kommentarer i koden)

Tips & krav til løsningen.:

a) Et viktig valg er hvilke klasser du har i programmet. I 'mønsterløsningen' er det følgende klasser: **Meteorologi, MetInst, Stasjon, Maaned, Dag** .

Slik begynner mønsterløsningen (med Javadoc kommentarer):

```
import easyIO.*;

/**
 * Omsluttende klasse for problemet, tar opp parametere
 * fra kommandolinja og starter kommandoløkkka. Feilmelding
 * hvis ikke minst to parametere.
 *****/
class Meteorologi{
    /**
     * Sjekker parametere, starter opp ordreløkkka etter at
     * filene er lest via konstruktøren til 'MetInst'
     *****/
    public static void main(String[] args) {

        if (args.length >= 2) {
            MetInst m = new MetInst(args[0],args[1]);
            m.ordreløkke();
        } else {
            System.out.println("Bruk: >java Oblig4 <fil med Stasjonsdata> < fil med Observasjonsdata>");
        }
    }
} // end class Oblig 4
```

b) Lag tre hjelpemetoder i den klassen som representerer måledata for en måned for en stasjon. Den første regner ut gjennomsnittlig nedbør per dag den måneden; den andre metoden regner ut gjennomsnittlig temperatur for måneden ut fra middeltemperaturene for hver dag, og den siste metoden teller opp antall uværstøgn..

c) Hvis du vil lage en nøkkel til en HashMap av noe som egentlig er et heltall, si: **int num;**, så lager du enkelt en nøkkel slik:

```
String s = num+"";
```

(dette lager en String med tallverdien i num ved å legge til den tomme tekststrengen)

d) En måte å behandle manglende data, er at alle -999'ene leses inn som om dette er virkelige data. De metoder som regner ut gjennomsnitt ol, må da for hvert tall sjekke om verdi \neq -999 og ellers ikke ta den med i beregningene. Slike metoder kan også selv returnere -999 dersom det ikke finnes **noen**

PUBLISERT FREDAG 26.10 (diff fra foreløpig versjon: korrektur og tips).
Rettelse 28.10: endret tips h)

data å gjøre beregningene på (enten f.eks. for at brukeren spesifiserer en måned vi ikke har data for - f.eks. måned: 10 – oktober, eller at **alle** data for vedkommende måned mangler). Husk at du samtidig må telle opp hvor mange dager du har virkelige observasjoner for, slik at gjennomsnittet blir riktig.

e) For å få de to filnavnene som parametre til 'main':

nevner du dem bare på samme linje som du starter programmet ditt. Anta at klassen din som inneholder main heter 'Meteorologi'. Du starter da programmet som følger:

```
>java Meteorologi stasjoner_norge.txt klima_data2012.txt
```

f) Hvis du leser inn data og så skriver dem ut på en PC via Windows, så vær klar over at æ, ø, å, Æ, Ø, Å blir skrevet ut som andre tegn på skjermen, og hvis man for eksempel taster inn æ, ø, å på en Windows-pc, så vil de ikke bli oppfattet som de æ-ene, ø-ene og å-ene du har lest inn fra fila. *Bruk derfor stasjonsnummerene* til å identifisere stasjonene, både i brukerdialogen og innad i programmet når du leser klimadata-filen. (eksempler på 'mishandling' av ÆØÅ er: TROMSĪ(90450)). Skriv derfor både ut navn og nummer på stasjonene når bruker skal velge stasjon. Da kan brukeren taste inn et stasjonsnummer når stasjonen velges.

g) Hvis du får behov for å lage gjennomsnitt av f.eks nedbør fra 2 eller flere måneder som har ett ulikt antall dager i seg med virkelige observasjoner, så ikke lag noe problem ut av det. La slike data for alle månedene telle likt i et større gjennomsnitt (del bare på antall måneder).

h) Eksempel på opprettelse av array med pekere til HashMap-er (rettet fra versjon 26.10):

```
HashMap<String, Person>[] personer =  
    (HashMap<String, Person>[]) new HashMap<?, ?>[10];
```

i) Hvis du har lyst til bare å nytte engelske variabelnavn, metodenavn og dokumentasjon, så gjør gjerne det, men brukerdialogen skal være på norsk.

j) Når du skal lage javadoc av systemet ditt (og vi antar at de eneste .java – filene på det filområdet er de som hører til Oblig4), så gir du kommandoen:

```
>javadoc -package *.java
```

Grunnen til å ha med parameteren **-package** er at den gjør at alle variable, klasser og metoder som det ikke står noen modifikator foran samt de det står public foran blir med i dokumentasjonen. Har du ikke med **-package**, vil bare de som er nevnt som **public** bli med i dokumentasjonen.

Husk også at du kan lage korte javadoc-kommentarer på en linje som f.eks:

```
/** Skriver ut bruker-menyvalg */  
void meny (Out ut) {  
    ...
```

Javadoc-kommentarer for en objektvariabel plasseres like over deklarasjonen, som f.eks:

```
/** Stasjonenes høyde over havet (meter) */  
int moh;
```

k) Når du lager filen "Resultat.txt" i oppgave 6, kan det være fornuftig å åpne den med append, dvs. f.eks.:

PUBLISERT FREDAG 26.10 (diff fra foreløpig versjon: korrektur og tips).
Rettelse 28.10: endret tips h)

```
Out res = new Out("Resultat.txt", true);
```

Gjør du det, vil alle resultatene fra testingen av Nordnorge mot Østlandet for alle månedene du prøver bli lagret på fila (du overskriver da ikke gamle data, men skriver nye resultater på slutten av fila).

D) Mønsterløsningen ble på ca. 420 linjer java-kode + 70 linjer med javadoc-kommentarer, men din løsning kan godt bli betydelig kortere eller lengre uten at den er noe dårligere for det.

Lykke til !