

Løsningsforslag til eksamen i INF1000

Are Magnus Bruaset
(oppgave 1a-e og 3)

Dag Langmyhr
(oppgave 1f-j og 2)

11. juni 2004

1 Flervalgsoppgave

(I løsningsforslaget her står noen kommentarer om hvorfor svaret er riktig. Dette er ikke forventet i eksamensbesvarelsene.)

1a

Riktig svar: **1**. Variabelen `tre` er en tekststreng satt til verdien "3"; derfor vil uttrykket `tre+4` evalueres som en tekst som er satt sammen av strengene "3" og "4".

1b

Riktig svar: **2**. Variabelen `tre` er et heltall satt til verdien 3. Uttrykket `4+tre` er derfor også et heltall, med verdi 7. Dette tallet skrives ut.

1c

Riktig svar: **5**. Programmet vil gå i evig løkke og skrive ut teksten "HEI" helt til det stoppes manuelt (med Control-C el. l.). Dette kommer av at while-løkka styres av variabelen `k` både direkte (`k<5`) og indirekte (forsett settes lik `false` når `k==3`), men samtidig oppdateres ikke `k` inne i løkka. Variabelen `k` vil dermed hele tida ha verdien 0 og variabelen `fortsett` vil hele tida ha verdien `true`.

1d

Programmet starter med en tom tekst `t`. I en for-løkke som gir `pos = 6, 5, 4, ..., 0` bygges teksten `t` opp ved å legge til delstrengen fra `s` som starter i posisjon `pos` og som er ett tegn langt. Det vil si at vi plukker ett og ett tegn bakfra i `s` og legger det til i `t`. Resultatet er en utskrift av teksten "ttakepa", altså "apekatt" baklengs.

1e

Programmet skriver ut teksten "B". Dette kommer av den første if-testen blir evaluert til `false` (fordi `n=3`, altså ikke mindre enn 5), samtidig som at if-testen i else-grenen til første test evalueres til `true` (fordi tegnet i posisjon 2 i teksten "hus" er 's', og det er nok at en av delene i testen er sann).

1f

Riktig svar: **6**. Det finnes ingen konstruktør for `Dato` med 2 parametre.

1g

Riktig svar: 5. Dato-konstruktøren i linje 8 (den uten parametre) vil sette d2. år til 2004, som så vil økes med 1 i linje 24.

1h

Riktig svar: 2. Elementet år vil nå bli en *klassevariabel*.

1i

Riktig svar: 4.

1j

Riktig svar: 3. Vi må bruke nøkkelen "påske" siden det er den som ble brukt da Dato-en ble lagt inn i HashMap-en. Resultatet fra get må typekonverteres til Dato.

2 Værdata

2a

UML-diagrammene er vist i figur 1 på neste side.

2b

Følgende kode finner nedbøren på Gardermoen 15. mars:

```
1   Værstasjon gmoen = (Værstasjon)met.stasjoner.get("GARDERMOEN");
2   double nb15mars = gmoen.mData[3-1].dData[15-1].nedbør;
```

2c

En enkel men oversiktlig måte å gjøre det på er å la klassene Meteorologisk, Værstasjon og Månedssdata ha hver sin metode mintemp som finner minimum blant sine data.

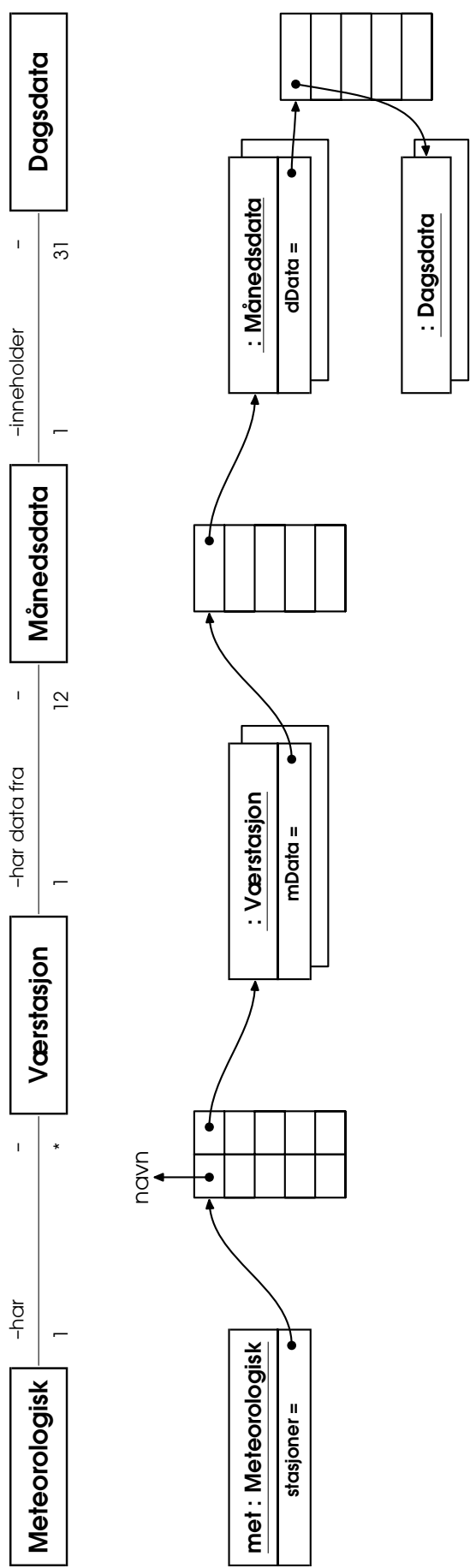
NB! Denne koden er litt for enkel til å være helt riktig: den antar at det finnes data for alle månedene i året og at alle måneder har 31 dager. Men en slik løsning vil bli godtatt til denne eksamenen.

Koden i main

```
double mintemp = met.mintemp();
```

Metoden mintemp i Meteorologisk

```
double mintemp () {
    double min = 999;
    Iterator it = stasjoner.keySet().iterator();
    while (it.hasNext()) {
        Værstasjon vs = (Værstasjon)it.next();
        double mx = vs.mintemp();
        if (mx < min) min = mx;
    }
}
```



Figur 1: UML klasse- og objekttdiagram til oppgave 2

```

        return min;
    }

```

Metoden mintemp i Værstasjon

```

double mintemp () {
    double min = 999;
    for (int m = 0; m < 12; ++m) {
        double mx = mData[m].mintemp();
        if (mx < min) min = mx;
    }
    return min;
}

```

Metoden mintemp i Månedssdata

```

double mintemp () {
    double min = 999;
    for (int d = 0; d < 31; ++d) {
        double mx = mData[d].temp_min;
        if (mx < min) min = mx;
    }
    return min;
}

```

3 Personnummer

3a

Observer at arrayene a og b i klassen Fødselsnummer inneholder vektene som brukes til å beregne summene sum1 og sum2. I tillegg har du en metode som sjekker om et tall er delelig med 11 eller ikke. Metoden gyldig kan dermed skrives slik:

```

1     boolean gyldig () {
2         // Tester om fødselsnummeret som er lagret i arrayen s er
3         // gyldig. Hvis det er tilfelle returneres verdien true, ellers
4         // returneres verdien false.
5
6         int sum1 = 0;
7         int sum2 = 0;
8         int i;
9         for (i = 0; i < 9; ++i)
10            sum1 += a[i]*s[i];
11        for (i = 0; i < 10; ++i)
12            sum2 += b[i]*s[i];
13
14        boolean ok = delelig11(sum1+s[9]);
15        ok = ok && delelig11(sum2+s[10]);
16
17        return ok;
18    }

```

3b

Vi må bygge opp en tekst bestående av sifrene lagret i arrayen s. Dette kan gjøres på flere måter, men den enkleste er å benytte pluss-operatoren til konkatenering av en tekst med et heltall (som konverteres til tekst). Metoden tekst kan skrives slik:

```

1 String tekst (boolean bare_dato) {
2     // Gjør om sifre som er lagret i arrayen s til
3     // en tekststreng og returnerer denne. Hvis bare_dato
4     // har verdien true returneres bare fødselsdato (s[0],...,s[5]),
5     // ellers returneres alle sifrene s[0],...,s[10], med et blankt
6     // tegn satt inn mellom fødselsdato og personnummer.
7
8
9     String t = ""; // Tom tekst for å kunne konkatenerer med +
10    int i;
11    for (i = 0; i < 6; ++i)
12        t += s[i];
13
14    if (!bare_dato) {
15        t += " ";
16        for (i = 6; i < 11; ++i)
17            t += s[i];
18    }
19    return t;
20 }

```