

1b

Vi har et program med følgende setninger:

```
int tre = 3;
System.out.println(4+tre);
```

Hva skrives ut når programmet kjøres?

1. 43
2. 7
3. 4tre
4. Ingen av alternativene ovenfor

1c

Vi har et program med følgende setninger:

```
int k = 0;
boolean fortsett = true;
while (k < 5 && fortsett) {
    if (k == 3) fortsett = false;
    System.out.println("HEI");
}
```

Hvor mange ganger skrives teksten "HEI" ut?

1. 0
2. 3
3. 4
4. 5
5. Ingen av alternativene ovenfor

1d

Vi har et program med følgende setninger:

```
String s = "apekatt";
String t = "";
int n = s.length();
for (int pos = n-1; pos >= 0; --pos)
    t += s.substring(pos,pos+1);
System.out.println(t);
```

Hva skjer når programmet kjøres?

(Fortsettes på side 3.)

1e

Vi har et program med følgende setninger:

```
String s = "hus";
int j = 34;
int n = s.length();

if (n > 5 && j == 34) {
    if (j < 40)
        System.out.println("A");
} else
    if (s.charAt(2) == 's' || j > 40)
        System.out.println("B");
    else
        System.out.println("D");
```

Hva skrives ut når programmet kjøres?

De følgende spørsmålene besvares med utgangspunkt i Java-koden vist i figur 1 på neste side.

1f

Hvilken verdi får d1 . år etter definisjonen i linje 21?

- 1
- 11
- 2001
- 2004
- Noe annet
- Definisjonen er ulovlig

1g

Hvilken verdi får d2 . år etter at ++-operasjonen i linje 24 er utført?

- 0
- 1
- 2000
- 2004
- 2005
- Noe annet
- Operasjonen er ulovlig

(Fortsettes på side 4.)

```
1 import easyIO.*;
2 import java.util.*;
3
4 class Dato {
5     private int dag, måned;
6     public int år;
7
8     Dato () {
9         dag = 1; måned = 1; år = 2004;
10    }
11
12    Dato (int d, int m, int å) {
13        dag = d; måned = m; år = å;
14    }
15
16    Dato les (In fil) {
17        return ...
18    }
19
20    public static void main (String arg[]) {
21        Dato d1 = new Dato(11, 6);
22
23        Dato d2 = new Dato();
24        ++d2.år;
25
26        Dato d3 = Dato.les(new In());
27
28        HashMap h = new HashMap();
29        h.put("nyttår", new Dato());
30        h.put("påske", new Dato(11,4,2004));
31
32    }
33 }
```

Figur 1: Klassen Dato

1h

Hva skjer hvis vi setter en `static` foran deklarasjonen av `år` i linje 6?

1. Ingen forskjell
2. Alle datoer vi lagrer vil ha samme år
3. Alle datoer vi lagrer vil være like
4. Endringen er ulovlig

1i

I metoden `les` ønsker vi å lese tre heltall som utgjør en dato fra filen `fil`. Hva er lovlige `return`-setninger i linje 17 for å få dette til?

(Fortsettes på side 5.)

1. `return new Dato();`
2. `return (Dato)new Dato();`
3. `return fil.inDato();`
4. `return new Dato(fil.inInt(), fil.inInt(), fil.inInt());`
5. `return new Dato(3*new inInt());`
6. `return Dato.in(fil);`
7. `return (Dato)fil.inText();`

1j

Hva er korrekte uttrykk for å få tak i datoen for påskedagen etter at linje 30 er utført?

1. `h.get("påske")`
2. `h.get("PÅSKE")`
3. `(Dato)h.get("påske")`
4. `(Dato)h.get("PÅSKE")`
5. `(String)h.get("påske")`
6. `(String)h.get("PÅSKE")`
7. `(Bil)h.get("påske")`
8. `(Bil)h.get("PÅSKE")`

Oppgave 2 Værdata (vekt 35%)

Anta at vi har to filer med værdata på samme format som i den siste obligatoriske oppgave.

Det er mange måter å lagre dataene på; i denne oppgaven antar vi at det er gjort slik det er vist i figur 2 på neste side.

2a UML

Tegn UML klassediagram og objektdiagram for datastrukturen.

2b Finn data

Skriv Java-kode (som skal ligge i `main`) for å finne nedbøren den 15.3 ved stasjon **GARDERMOEN**.

2c Finn kaldeste observasjon

Skriv Java-kode (som skal ligge i `main`) som finner den laveste temperaturen som er observert og skriver den ut.

(Fortsettes på side 6.)

```
import java.util.*;

class Meteorologisk {
    HashMap stasjoner = new HashMap();

    public static void main (String arg[]) {
        Meteorologisk met = new Meteorologisk();
        // ...
    }
}

class Værstasjon {
    int nummer, hoh;
    String navn, kommune, fylke;
    boolean østlandet, vestlandet;
    Månedldata mData[] = new Månedldata[12];
    //...
}

class Månedldata {
    int månedsNr;
    Dagsdata dData[] = new Dagsdata[31];
    //...
}

class Dagsdata {
    double vind, nedbør, temp_min, temp_max;
    //...
}
```

Figur 2: Datastruktur for meteorologiske data

Oppgave 3 Personnummer (vekt 35%)

Fra og med 1964 har hver person med fast bosted i Norge fått tildelt et 11-sifret **fødselsnummer**. De seks første sifrene angir fødselsdato på formatet **ddmmåå**, der **dd**, **mm** og **åå** angir posisjoner for sifrene i henholdsvis dag, måned og år. Bare de to siste sifrene av årstallet brukes. De fem siste sifrene i fødselsnummeret kalles **personnummer**.

Et personnummer er delt inn i to grupper. De tre første sifrene angir et individnummer som skal skille personer født på samme dato. Det tredje av disse sifrene vil også angi kjønn, ved at kvinner får tildelt et partall, mens menn får tildelt et oddetall. De to siste sifrene i personnummeret er **kontrollsifre** som beregnes ut fra alle foregående siffer i fødselsnummeret.

Eksempel Betrakt fødselsnummeret

(Fortsettes på side 7.)

```
class Fødselsnummer {  
  
    // Lagring av fødselsnummer, siffer for siffer  
    int[] s = new int[11];  
  
    // Faktorer for kontroll av fødselsnummer  
    int[] a = {3,7,6,1,8,9,4,5,2};  
    int[] b = {5,4,3,2,7,6,5,4,3,2};  
  
    Fødselsnummer (String fnr) {  
        // Kontrollerer at fnr er en tekststreng bestående av 11 siffer.  
        // Deler opp strengen siffer for siffer. Lagrer hvert siffer  
        // som et heltall i arrayen s slik at det første sifferet ligger  
        // i s[0], og det siste i s[10].  
  
        // Du kan anta at denne metoden allerede finnes og oppfører  
        // seg som beskrevet ovenfor. Denne metoden skal derfor  
        // ikke skrives.  
  
        ...  
    }  
  
    boolean delelig11 (int tall) {  
        // Tester om fødselsnummeret er delelig med 11. Hvis det  
        // er tilfelle returneres verdien true, ellers returneres  
        // verdien false.  
  
        int tmp = tall/11;  
        return (11 * tmp == tall);  
    }  
  
    boolean gyldig () {  
        // Tester om fødselsnummeret som er lagret i arrayen s er gyldig.  
        // Hvis det er tilfelle returneres verdien true,  
        // ellers returneres verdien false.  
  
        ...  
    }  
  
    String tekst (boolean bare_dato) {  
        // Gjør om sifre som er lagret i arrayen s til  
        // en tekststreng og returnerer denne.  
  
        ...  
    }  
}
```

Figur 3: Klassen Fødselsnummer

(Fortsettes på side 8.)

081273 24604

Dette tilhører en person som er født 8. desember 1973. Siden det niende sifferet (6) er et partall, er denne personen en kvinne. Kontrollsifrene i de to siste posisjonene, 0 og 4, er beregnet ut fra de foregående sifrene.

I denne oppgaven skal du skrive innmaten i to metoder som inngår i en klasse `Fødselsnummer` som er vist i figur 3 på forrige side.

3a

For å unngå feil ved inntasting av fødselsnummere i et datasystem er det nødvendig å kontrollere om de inntastede sifrene er gyldige. Du skal gjøre slik kontroll tilgjengelig i klassen ovenfor ved å skrive innholdet i metoden `gyldig`. Algoritmen for sjekking av kontrollsifrene er som følger:

1. Beregn summen

$$sum1 = 3 * s[0] + 7 * s[1] + 6 * s[2] + \dots + 5 * s[7] + 2 * s[8]$$

der de ni første sifrene i fødselsnummeret hentes fra arrayen `s` deklartert i klassen `Fødselsnummer`. I et gyldig fødselsnummer skal summen av `sum1` og det første kontrollsifferet (`s[9]`) være delelig med 11. Benytt arrayen `a` og metoden `delelig11` som begge er definert ovenfor.

2. Beregn summen

$$sum2 = 5 * s[0] + 4 * s[1] + 3 * s[2] + \dots + 3 * s[8] + 2 * s[9]$$

der de ti første sifrene i fødselsnummeret er hentet fra arrayen `s` deklartert ovenfor. I et gyldig fødselsnummer skal summen av `sum2` og det andre kontrollsifferet (`s[10]`) være delelig med 11. Benytt arrayen `b` og metoden `delelig11` som begge er definert ovenfor.

Begge kontrollsifrene må være gyldige for at fødselsnummeret skal aksepteres.

3b

I forbindelse med utskrifter er det hensiktsmessig med en tekstlig representasjon av et fødselsnummer. Klassen `Fødselsnummer` har derfor en metode `tekst(boolean bare_dato)` som kan gjøre om sifrene i arrayen `s` til en tekststreng og returnere denne. Hvis parameteren `bare_dato` har verdien `true`, returneres kun fødselsdatoen (seks siffer), ellers returneres hele fødselsnummeret (11 siffer). I det siste tilfellet skal fødselsdatoen og personnummeret separeres ved å sette inn et blankt tegn i teksten som returneres.

Skriv innholdet i metoden `tekst`.