

## INF1000 (Uke 13)

### Resten av eksamen H03 + del av V05

---

Grunnkurs i programmering  
Institutt for Informatikk  
Universitet i Oslo

Are Magnus Bruaset og Arild Waaler



## Eksamen H03, oppgave 2

---

- Oppgave 1 ble forelest 23. april, se slides på hjemmesiden

2007-04-30

2



## Oppgave 2.1

---

- Metoden nedenfor har to parametre: en heltallsarray **a** og et heltall **k**.

Metoden skal telle opp hvor mange verdier i **a** som er lik **k**, og skal returnere dette antallet.

Fyll inn innholdet i metoden:

```
int finnAntall(int[] a, int k) { ... }
```

2007-04-30

3



## Oppgave 2.1 forts.

---

```
int finnAntall(int[] a, int k) {  
    int ant = 0;  
  
    for (int i = 0; i < a.length; ++i)  
        if (a[i] == k) ++ant;  
  
    return ant;  
}
```

2007-04-30

4



## Oppgave 2.2

- Du lager et datasystem for en idrettsklubb. Systemet skal inneholde følgende informasjon om medlemmene:

Navn, adresse, fødselsnummer (11 siffer), telefonnummer, statsborgerskap, yrke og sivil stand

Kan et slikt system tas i bruk uten konsesjon?

- Ja
- Nei, fordi følgende paragrafer i personvernet er relevante: ...

2007-04-30

5



## Oppgave 2.2 forts.

- De mest sentrale paragrafene:
  - §8 Det skal være saklig grunn for registeret.
  - §9 Sensitive opplysninger kun hvis den registrerte samtykker. §2 forteller hva som er sensitivt.
  - §19 Den registrerte skal vite om registeret.
  - §33 Man må ha konsesjon når det lagres sensitive opplysninger.
- I dette registeret har vi: Fødsels- og personnummeret regnes som sensitivt. Statsborgerskap, yrke og sivilstand er muligens sensitive opplysninger (og neppe aktuelle for et idrettslag.)
- Konklusjon:
  - Konsesjon er nødvendig pga §8, §9 og §33.

2007-04-30

6



## Oppgave 2.3

- Metoden nedenfor skal skrive ut verdiene til de to parametrene på hver sin linje slik at den som er alfabetisk først kommer på første linje, og den som er alfabetisk sist kommer på neste linje.

Fyll inn innholdet i metoden:

```
void skrivUtSortert(String tekst1, String tekst2) {...}
```

2007-04-30

7



## Løsning oppgave 2.3

```
void skrivUtSortert(String tekst1, String tekst2) {  
  
    String s1, s2;  
  
    if (tekst2.compareTo(tekst1) > 0) {  
        s1 = tekst1;  
        s2 = tekst2;  
    } else {  
        s2 = tekst1;  
        s1 = tekst2;  
    }  
  
    System.out.println(s1);  
    System.out.println(s2);  
  
}
```

2007-04-30

8



## Oppgave 2.4

- Du får et uferdig program for å holde orden på nettsteder. For hvert nettsted skal programmet lagre navnet på nettstedet (f.eks. "Hjemmesiden til INF1000"), nettadressen (f.eks. "http://www.ifi.uio.no/inf1000"), og en tekstlig beskrivelse av nettstedet (f.eks. "Dette nettstedet er nyttig for alle som følger kurset INF1000").
- Programmet skal ha funksjonalitet for å legge inn informasjon om et nytt nettsted, fjerne informasjon om et nettsted, og skrive ut en oversikt over alle lagrede nettsteder.

2007-04-30

9



## Oppgave 2.4

Gitt følgende program:

```
import easyIO.*;
import java.util.*;

class NettstedProg {
    public static void main (String[] arg) {
        NettstedOversikt no = new NettstedOversikt();
        no.ordreløkke();
    }
}

class NettstedOversikt {
    In tastatur = new In();
    HashMap nettsteder = new HashMap();

    int lesOrdre () {
        int k = 0;
        while (k < 1 || k > 4) {
            System.out.println("1: Legg inn nytt nettsted");
            System.out.println("2: Fjern nettsted");
            System.out.println("3: Lag oversikt over nettsteder");
            System.out.println("4: Avslutt");
            k = tastatur.inInt();
        }
        return k;
    }
}
```

2007-04-30

10



## Oppgave 2.4 forts.

```
void ordreløkke () {
    boolean fortsett = true;
    while (fortsett) {
        switch (lesOrdre()) {
            case 1: leggInnSted(); break;
            case 2: fjernSted(); break;
            case 3: lagOversikt(); break;
            case 4: fortsett = false;
        }
    }
}

void leggInnSted () {
    // Les navn, nettadresse og beskrivelse fra terminal,
    // opprett nytt Nettsted-objekt og legg det inn i
    // HashMap-en 'nettsteder' med adressen som nøkkel.
}
```

2007-04-30

11



## Oppgave 2.4 forts.

```
void fjernSted() {
    // Les en tekststreng fra terminal. Gå så gjennom alle
    // nøklene i HashMap-en 'nettsteder' for å finne alle de
    // nettstedene med nettadresse som enten inneholder
    // tekststrengen eller er identisk med tekststrengen.
    // For hvert slikt nettsted skal programmet spørre
    // brukeren om det skal slettes, og hvis brukeren svarer
    // ja, så fjernes nettstedet fra HashMap-en 'nettsteder'.
    // Lese f eks tekststrengen "uio.no", så skal programmet
    // i tur og orden presentere for brukeren hvert nettsted
    // med "uio.no" i sin nettadresse og spørre om
    // nettstedet skal fjernes fra registeret.
}

void lagOversikt () {
    // Løp gjennom alle nettstedene som er lagret og skriv
    // ut innholdet av hver dem.
}

} // Slutt på class NettstedOversikt
```

2007-04-30

12



## Oppgave 2.4 forts.

```
class Nettsted {
    private String navn;
    private String nettadresse;
    private String beskrivelse;

    Nettsted (String navn, String nettadresse,
              String beskrivelse) {
        this.navn = navn;
        this.nettadresse = nettadresse;
        this.beskrivelse = beskrivelse;
    }

    void skrivUt () {
        System.out.println("Navn: "+navn);
        System.out.println("Nettadresse: "+nettadresse);
        System.out.println("Beskrivelse: "+beskrivelse);
    }
} // Slutt på class Nettsted
```

2007-04-30

13



## Oppgave 2.4 (A)

Gjør ferdig følgende:

```
void leggInnSted () {

    // Les navn, nettadresse og beskrivelse fra terminal,
    // opprett nytt Nettsted-objekt og legg det inn I
    // HashMap-en 'nettsteder' med adressen som nøkkel.

    System.out.print("Navn: ");
    String navn = tastatur.inWord("\n");
    System.out.print("Adresse: ");
    String adresse = tastatur.inWord("\n");
    System.out.print("Beskrivelse: ");
    String beskr = tastatur.inWord("\n");

    ...
}
```

2007-04-30

14



## Løsning oppgave 2.4 (A)

```
void leggInnSted () {

    // Les navn, nettadresse og beskrivelse fra terminal,
    // opprett nytt Nettsted-objekt og legg det inn I
    // HashMap-en 'nettsteder' med adressen som nøkkel.

    System.out.print("Navn: ");
    String navn = tastatur.inWord("\n");
    System.out.print("Adresse: ");
    String adresse = tastatur.inWord("\n");
    System.out.print("Beskrivelse: ");
    String beskr = tastatur.inWord("\n");

    nettsteder.put(adresse, new Nettsted(navn, adresse, beskr));

}
```

2007-04-30

15



## Oppgave 2.4 (B)

- Venter litt med denne og tar punkt (C) først

2007-04-30

16



## Oppgave 2.4 (C)

Gjør ferdig følgende:

```
void lagOversikt () {
    // Løp gjennom alle nettstedene som er lagret og skriv
    // ut innholdet av hver dem.
}
```

2007-04-30

17



## Løsning oppgave 2.4 (C)

```
void lagOversikt () {
    // Løp gjennom alle nettstedene som er lagret og skriv
    // ut innholdet av hver dem.

    Iterator it = nettsted.values().iterator();
    while (it.hasNext()) {
        Nettsted ns = (Nettsted)it.next();
        ns.skrivUt();
    }
}
```

2007-04-30

18



## Oppgave 2.4 (B)

Gjør ferdig følgende:

```
void fjernSted() {
    // Les en tekststreng fra terminal. Gå så gjennom alle
    // nøklene i HashMap-en 'nettsteder' for å finne alle de
    // nettstedene med nettadresse som enten inneholder
    // tekststrengen eller er identisk med tekststrengen.
    // For hvert slikt nettsted skal programmet spørre
    // brukeren om det skal slettes, og hvis brukeren svarer
    // ja, så fjernes nettstedet fra HashMap-en 'nettsteder'.
    // Les f eks tekststrengen "uio.no", så skal programmet
    // i tur og orden presentere for brukeren hvert nettsted
    // med "uio.no" i sin nettadresse og spørre om
    // nettstedet skal fjernes fra registeret.

    System.out.print("Adresse: ");
    String adresse = tastatur.inWord("\n");

    ...
}
```

2007-04-30

19



## Første spørsmål

- Det første spørsmålet man stiller seg når man skal programmere, er:

*"Hva skal gjøres?"*

- Her er svaret:

Spør brukeren om en nettadresse. Gå gjennom alle nettsteder og fjern de som matcher.

2007-04-30

20



## Andre spørsmål

- Neste spørsmål er:

“*Hvordan* skal jeg få det til?”

- Be brukeren om en nettsadresse:  
Trivielt
- Gå gjennom alle nettstedene:  
Vanlig opplegg med en `Iterator`
- Hvordan sjekke hvilke som matcher:  
Med String-metoden `indexOf`
- Hvordan fjerne en post i `HashMap`-en:  
Med metoden `remove`

2007-04-30

21



## Løsning oppgave 2.4 (B)

```
void fjernSted() {
    System.out.print("Adresse: ");
    String adresse = tastatur.inWord("\n");
    Iterator it = nettstedet.keySet().iterator();

    while (it.hasNext()) {
        String adr = (String)it.next();
        if (adr.indexOf(adresse) >= 0) {
            System.out.print("Skal "+adr+" fjernes? ");
            String svar = tastatur.inWord();
            if (svar.charAt(0)=='j' || svar.charAt(0)=='J')
                it.remove();
        }
    }
}
```

2007-04-30

22



## Oppgave 2.5

- (*Denne oppgaven bør tas til slutt.*) Metoden nedenfor skal – uten bruk av array – skrive ut verdiene til de tre parametrene på hver sin linje i stigende rekkefølge. Metoden skal først bytte om på verdiene i `a`, `b` og `c` slik at  $a \leq b \leq c$ , og deretter skrives innholdet av `a`, `b` og `c` ut (i den rekkefølgen, og på hver sin linje).

Fyll inn innholdet i metoden:

```
void skrivUtSortert2 (int a, int b, int c) { }
```

2007-04-30

23



## Løsning oppgave 2.5

- Dette er en oppgave som ikke er helt triviell.

Man må sjekke to og to verdier mot med hverandre og så bytte dem om hvis de står i gal rekkefølge.

2007-04-30

24

## Løsning oppgave 2.5 forts.

```
void SkrivUtSortert2 (int a, int b, int c) {
    int tmp;

    if (a > b) {
        // Bytt om a og b:
        tmp = a; a = b; b = tmp;
    }
    // Nå vet vi at a<=b.

    if (b > c) {
        // Bytt om b og c:
        tmp = b; b = c; c = tmp;
    }
    // Nå vet vi at a<=c && b<=c.

    if (a > b) {
        // Bytt om a og b:
        tmp = a; a = b; b = tmp;
    }

    System.out.println(a);
    System.out.println(b);
    System.out.println(c);
}
```

2007-04-30

25

## Eksamen V05

2007-04-30

26

## Oppgave 1 - Kortsvarsoppgave

(a)

```
class SkrivHei {
    public static void main (String[] args) {
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 2; j ++ ) {
                i = i + 4;
                System.out.println("hei");
            }
        }
    }
}
```

Hvor mange ganger blir teksten hei skrevet ut når programmet SkrivHei kjøres?

Svar: 2 ganger

2007-04-30

27

## Oppgave 1 (b)

Hva skrives ut på skjermen når programmet nedenfor kjøres?

```
class EnkelRegning {
    public static void main(String[] args) {
        int a = 4;
        int b = 9;
        a = b;
        b++;
        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
}
```

Svar: a = 9, b=10

2007-04-30

28

## Oppgave 1 (c)

```
class SkrivUt2{
    public static void main (String[] args){
        String s = "abcdefghijklm";
        String t = s.substring(0,3);
        System.out.println("t er nå:" + t);
        int lengde = s.length();
        int kvart = lengde/4;
        int halv = lengde/2;
        String u = s.substring(halv - kvart, halv + kvart);
        System.out.println("u er nå:" + u);
        if (s.indexOf("deF")>(-1)) {
            System.out.println("deF finnes");
        } else {
            System.out.println("deF finnes ikke");
        }
        if (s.endsWith("klm")){
            System.out.println("s ender med klm");
        }
    }
}
```

Hva skrives ut på skjermen?

Svar:

t er nå:abc

u er nå:defghi

deF finnes ikke

s ender med klm

2007-04-30

29

## Oppg. 1 (d)

```
class Beregninger{
    public static void main(String[] args){
        int x = 9;
        int y = 3;
        int z = x - y;
        System.out.println("verdien til z er " + z);
        if ((x-z) == y){
            System.out.println("like");
        } else {
            System.out.println("ulike");
        }
        if ((x > 8) || (y < 2)){
            System.out.println("sann");
        } else {
            System.out.println("usann");
        }
        x++;
        y += x;
        int d = --x + y--;
        System.out.println("d = " + d);
    }
}
```

Hva skrives ut på skjermen?

Svar:

verdien til z er 6

like

sann

d = 22

2007-04-30

30

## Oppgave 1 (e)

```
class WhileTest {
    public static void main (String[] args)
    {
        boolean fortsett = true;
        int k = 3;
        while (fortsett) {
            k += 3;
            fortsett = !fortsett;
            System.out.println("k = " + k);
        }
    }
}
```

Hva skrives ut på skjermen?

Svar:

k=6

2007-04-30

31

## Oppgave 1 (f)

```
int[] a = new int[50];
... // fyller inn tall i arrayen a
int s = 1;
for (int i = 0; i < a.length; ++i) {
    s *= a[i];
}
```

Hva blir innholdet i variabelen s etter at denne koden er utført?

Svar:

produktet av alle verdiene i array a

2007-04-30

32



## Oppgave 1 (g)

```
1 class Person [  
2     int navn;  
3     int alder;  
4     String adresse;  
5  
6     public Person (String n, int a, String adr) {  
7         navn = n;  
8         alder = a;  
9         adresse = adr; Svar:  
10    }  
11 ]
```

Finne og korriger alle feil i denne koden.

- Linje 1 og 11 feil bruk av parentes.
- Linje 2 navn er deklartert som en int, men i linje 7 blir navn satt lik en String.

2007-04-30

33

## Oppgave 2 – class Geografikunnskaper

```
import java.util.*;  
class Geografikunnskaper {  
    HashMap LandOversikt = new HashMap();  
  
    void leggTilLand(String navn, int innbyggertall) {  
        Land land = new Land(innbyggertall);  
        LandOversikt.put(navn, land);  
    }  
  
    void leggTilByILand(String navnLand, String navnBy,  
                        int innbyggertall, boolean hovedstad) {  
        boolean finnes = LandOversikt.containsKey(navnLand);  
        if (!finnes) {  
            Land land = new Land(0);  
            LandOversikt.put(navnLand, land);  
        }  
        Land nland = (Land) LandOversikt.get(navnLand);  
        nland.leggTilBy(navnBy, innbyggertall, hovedstad);  
    }  
}
```

2007-04-30

34

## Oppgave 2 – class Geografikunnskaper

```
public static void main(String args[]) {  
    Geografikunnskaper geo = new Geografikunnskaper();  
    geo.leggTilLand("Norge", 4525000);  
    geo.leggTilLand("Sverige", 9040000 );  
    geo.leggTilByILand("Norge", "Oslo", 530000, true);  
    geo.leggTilByILand("Norge", "Moss", 26000, false);  
    geo.leggTilByILand("Norge", "Narvik", 19000, false);  
    geo.leggTilByILand("Norge", "Hamar", 26500, false);  
    // Anta at programmet senere skal utvides slik at elever  
    // kan legge inn informasjon om nye land og byer ved bruk  
    // av meny og inntasting på skjermen. NB! Dette skal ikke  
    // lages nå.  
    // Her skal du sørge for at alle byer i alle land blir  
    // skrevet ut på skjermen.  
}  
} // Slutt på class Geografikunnskaper
```

2007-04-30

35

## Oppgave 2 – class Land

```
class Land{  
    int innbyggertall;  
    HashMap Byer = new HashMap();  
  
    Land(int innbyggertall) {  
        this.innbyggertall = innbyggertall;  
    }  
  
    void leggTilBy(String navn, int innbyggertall,  
                  boolean hovedstad) {  
        By by = new By(innbyggertall, hovedstad);  
        Byer.put(navn, by);  
    }  
  
    void skrivUtAlleLand() {  
        // Denne metoden skal du skrive i oppgave 2b  
    }  
} // Slutt på class Land
```

2007-04-30

36

## Oppgave 2 – class By

```
class By{
    int innbyggertall;
    boolean hovedstad;

    By(int innbyggertall, boolean hovedstad) {
        this.innbyggertall = innbyggertall;
        this.hovedstad = hovedstad;
    }

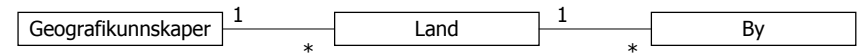
    void skrivUt() {
        // Denne metoden skal du skrive i oppgave 2b
    }
} // Slutt på class By
```

2007-04-30

37

## Oppgave 2 a

- Lag UML klassediagram for systemet

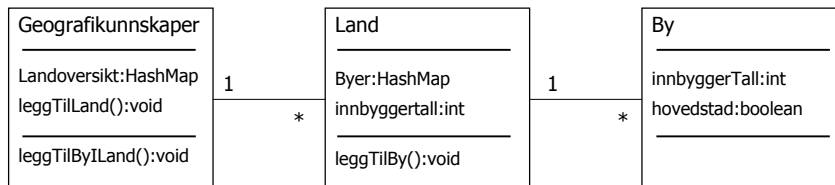


2007-04-30

38

## Oppgave 2 a

- Eller:



2007-04-30

39

## Oppgave 2b

- Lag en rutine som kan skrive ut:

```
Norge    4525000
*****
Oslo     530000
Moss     260000
Narvik   190000
Hamar    265000
*****
Sverige  9040000
```

- For land uten registrerte byer skrives kun landsinfo.

2007-04-30

40



## Oppgave 2 b – class By

```
void skrivUt(){
    System.out.println(innbyggertall);
}
```

2007-04-30

41



## Oppgave 2 b – class Land

```
void skrivUtAlleLand(){
    System.out.println(innbyggertall);
    System.out.println("*****");
    Iterator it = Byer.keySet().iterator();
    while (it.hasNext()){
        String nøkkel = (String) it.next();
        System.out.print(nøkkel + " ");
        By b = (By) Byer.get(nøkkel);
        b.skrivUt();
    }
}
```

2007-04-30

42



## Oppgave 2 b – class Geografikunnskaper

```
void skrivUt(){
    Iterator it = LandOversikt.keySet().iterator();
    while (it.hasNext()){
        String nøkkel = (String) it.next();
        System.out.println(nøkkel + " ");

        Land land = (Land) LandOversikt.get(nøkkel);
        land.skrivUtAlleLand();
        System.out.println("*****");
    }
}
```

2007-04-30

43



## Oppgave 2 b - main

```
geo.skrivUt();
```

2007-04-30

44

## Oppgave 3 – DNA-sekvenser

Gitt DNA-sekvensen AATGGATC.

Denne sekvensen består av 8 symboler, hvorav 3 forekomster av A.

Dermed er den relative frekvensen av A i sekvensen  $\frac{3}{8} = 0,375$ .

2007-04-30

45

## Oppgave 3 (a)

```
double[] symbolFrekvens(String sekvens){
    int a = 0;
    int t = 0;
    int c = 0;
    int g = 0;
    String SEKVENS = sekvens.toUpperCase();
    int n = SEKVENS.length();
    for (int i = 0; i < n; i++){
        char s = SEKVENS.charAt(i);
        switch (s) {
            case 'A': a++;
                       break;
            case 'T': t++;
                       break;
            case 'C': c++;
                       break;
            case 'G': g++;
        }
    }
    double[] frekvens = {(double) a/n, (double) t/n,
                        (double) c/n, (double) g/n};
    return frekvens;
}
```

2007-04-30

46

## Oppgave 3 (b)

Lag en klasse `DNasekvens` som inneholder følgende informasjon:

- navnet på sekvensen,
- selve sekvensen,
- lengden på sekvensen (antall symboler),
- frekvensen av A, T, C og G.

Lag en konstruktør i klassen slik at et nytt `DNasekvens` objekt kan opprettes ved kodesetningen

```
DNasekvens a = new DNasekvens(navn,sekvens);
```

der argumentene `navn` og `sekvens` begge er av type `String`.

2007-04-30

47

## Oppgave 3 (b)

```
class DNasekvens{
    String navn;
    String sekvens;
    double[] frek = new double[4];
    int lengde;

    double[] symbolFrekvens(String sekvens){
        // SE OPPGAVE 3A
    }

    DNasekvens(String navn, String sekvens){
        this.navn = navn;
        this.sekvens = sekvens.toUpperCase();
        lengde = sekvens.length();
        frek = symbolFrekvens(sekvens);
    }
}
```

2007-04-30

48



## Oppgave 3 (c)

Eksempel på filen DNA.txt.:

```
137
AY1231 AAATCAGAAG
AY5432 GGAATCCAGTAAAA
AY3234 GGAGTCGATGA
AY3423 GGAGTCGATCCCAGA
...
```

Lag en metode

```
DNasekvens[] lesSekvenserFraFil (String filnavn) {
    ...
}
```

som leser inn alle DNA-sekvensene i en array bestående av `DNasekvens`-objekter. Denne arrayen returneres fra metoden.



## Oppgave 3 (c)

```
DNasekvens[] lesSekvenserFraFil(String filnavn){
    In innfil = new In(filnavn);
    int ant = innfil.inInt();
    DNasekvens[] sekvensene = new DNasekvens[ant];
    for (int i = 0; i < ant; i++){
        String navn = innfil.inWord();
        String sek = innfil.inWord();
        sekvensene[i] = new DNasekvens(navn,sek);
    }
    return sekvensene;
}
```