

INF1000 (Uke 14)

Resten av eksamen H03 + del av V05

Grunnkurs i programmering
Institutt for Informatikk
Universitet i Oslo

Are Magnus Bruaset og Anja B. Kristoffersen



Eksamen H03, oppgave 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) { ... }
```



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;  
  
}
```



Oppgave 2.2

- Du lager et datasystem for en idrettsklubb. Systemet inneholder 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: ...



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.



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) {...}
```



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);  
  
}
```




Oppgave 2.4

- Du får et uferdig program for å holde orden på nettsteder (dvs internett-sider). 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.



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;
    }
}
```



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.
}
```



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'.
    // 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.
}

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

} // Slutt på class NettstedOversikt
```



Oppgave 2.4 forts.

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

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

    void skrivUt () {
        System.out.println("Navn: "+navn);
        System.out.println("Nettsadresse: "+nettsadresse);
        System.out.println("Beskrivelse: "+beskrivelse);
    }
} // slutt på class Nettsted
```



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");  
  
    ...  
}
```



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));  
  
}
```



Oppgave 2.4 (B)

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



Oppgave 2.4 (C)

Gjør ferdig følgende:

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



Løsning oppgave 2.4 (C)

```
void lagOversikt () {  
    // Løp gjennom alle nettstedene som er lagret og skriv  
    // ut innholdet av hver dem.  
  
    Iterator it = nettsteder.values().iterator();  
    while (it.hasNext()) {  
        Nettsted ns = (Nettsted)it.next();  
        ns.skrivUt();  
    }  
}
```



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");

    ...
}
```



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.



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`

Løsning oppgave 2.4 (B)

```
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.

    System.out.print("Adresse: ");
    String adresse = tastatur.inWord("\n");
    Iterator it = nettsted.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();
        }
    }
}
```



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) { }
```



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.



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);
}
```



Eksamen V05, oppgave 2

- Oppgave 1 og 3 blir drøftet på gruppene denne uken



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);
    }
}
```



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
```



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
```



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
```

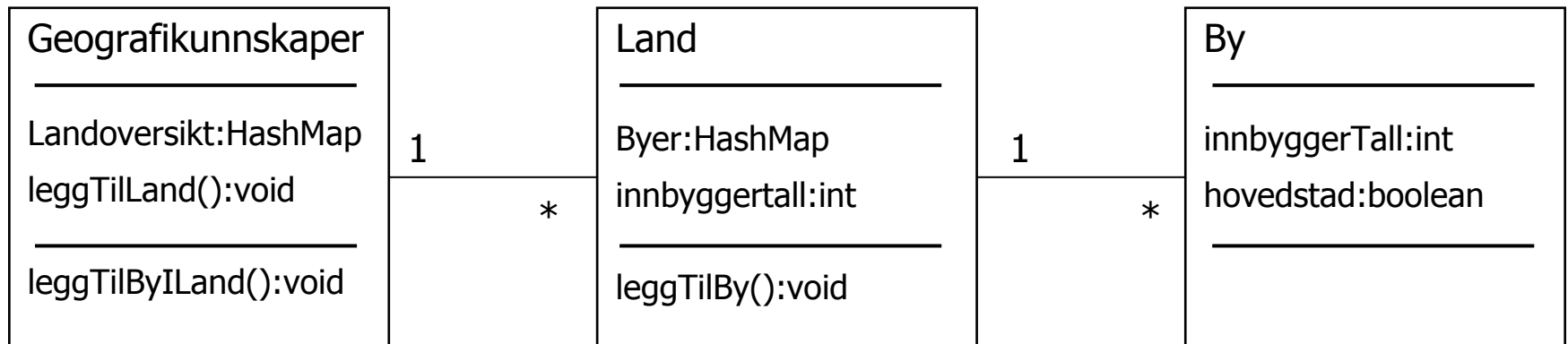
Oppgave 2 a

- Lag UML klassediagram for systemet



Oppgave 2 a

- Eller:





Oppgave 2b

- Lag en rutine som kan skrive ut:

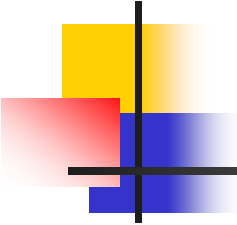
```
Norge      4525000
*****
Oslo       530000
Moss       26000
Narvik     19000
Hamar      26500
*****
Sverige    9040000
```

- For land uten registrerte byer skrives kun landsinfo.



Oppgave 2 b – class By

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



Oppgave 2 b – class Land

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



Oppgave 2 b – class Geografikunnskaper

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



Oppgave 2 b - main

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