

## INF1000 – våren 2009 Aller siste forelesning!

Svar på 2 spørsmål mottatt via e-post

### Spørsmål 1:

”Er det lov å lage register av offentlig tilgjengelig informasjon eller ikke? (typisk navn, tlfnr, inntekt og fødselsår henta fra ulike steder som eks skattelistingene) Klarte ikke helt å finne et svar jeg blei overbevist om at var rett...”

### ”Listesaken” 1977

- Politiet gikk i august 1977 til razzia mot Ny Tids redaksjonslokaler i Oslo og beslagla lister med 600 navn over ansatte i de hemmelige tjenestene
- SV-eren Ivar Johansen hadde samlet inn data fra åpne kilder som telefonkatalogen, Tjenestemannsbladet og Politibladet, og noterte bilnumrene til de ansatte
- Johansen ble dømt til delvis betinget fengsel i ett år, hvorav han sonet 60 dager

### Konklusjon

- Generelt: å lage et register i seg selv faller inn under personvernloven - dvs. det kan både være lovlig og ulovlig
- Eks: lage et register som inneholder mer enn det mitt firma har legitim rett til å vite - for eksempel data fra skattelistingene over alle medlemmer i en sportsklubb
- Loven gjelder selv om opplysningene du stapper inn er offentlig tilgjengelig (fra ulike kilder)

## Spørsmål 2:

”eksamensoppgave høst 2004 - oppgave 14:  
Fasiten setter public foran metodene som leser til/fra fil. Er dette nødvendig? hvorfor setter de public der, og ville det blitt riktig også uten noe foran?”

## Tilgangsmodifikatorer i Java

- I Java er det mulig å begrense tilgangen til metoder og variable i en klasse
- Dette gjøres med tilgangsmodifikatorer (eng: access modifiers)
- Inngående kjennskap til dette ikke pensum
- Men: bør kjenne til `private` og `public`

## `private` vs. `public`

- Hvis en variabel eller metode er `private`:
  - Kun tilgjengelig fra den samme klassen
- Hvis en variabel eller metode er `public`:
  - Tilgjengelig fra alle klasser i hele programmet
- Hvis verken `private` eller `public`:
  - Tilgjengelig fra alle klasser i samme *pakke* (= i samme katalog)
- Siden alle klasser i løsningsforslaget er i samme katalog, vil *ingen* modifikator tilsvare `public`

## Eksamen høst 2004 – oppg. 14:

Anta at du har en tekstfil Bok.txt som f.eks. inneholder teksten i en roman. Lag et komplett, kompilert program som leser filen ord for ord og som teller opp antall forekomster av hvert enkelt ord i filen. Programmet ditt skal deretter opprette en ny fil Ordforekomster.txt og skrive resultatet av analysen til denne filen. Følgende opplysninger om Bok.txt skal skrives til denne filen: (a) antall ord totalt; (b) antall ulike ord; og (c) en liste over alle ord som forekommer minst 6 ganger. Hver linje av denne listen skal inneholde ett ord og antall forekomster av dette ordet, atskilt av blanke tegn.

Du kan anta at filen Bok.txt kun inneholder bokstaver og blanke tegn, dvs spesialtegn som komma, punktum, kolon, osv, forekommer ikke i teksten. Du kan også anta at alle ord i teksten er skrevet med små bokstaver, slik at du slipper problemer med stor forbokstav i begynnelsen av en setning, osv.

Hint: bruk en HashMap til å holde rede på ordene du leser i Bok.txt og antall forekomster av dem. Tenk deg nøye om hva du skal legge inn i HashMap'en. Husk at både nøkkelen og verdien du legger inn i en HashMap må være (pekere til) objekter. Du vil kanskje ønske å definere en klasse Ord hvor hvert objekt lagrer et ord og antall forekomster av ordet.

```

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

class BokMain {
    public static void main (String[] args) {
        BokAnalyse ba = new BokAnalyse();
        ba.lesFraFil("Bok.txt");
        ba.skrivTilFil("Ordforekomster.txt");
    }
}

class Forekomster {
    private int antall = 1;
    public void nyForekomst() {
        antall++;
    }
    public int fåAntall() {
        return antall;
    }
}

class BokAnalyse {
    private HashMap alleOrd = new HashMap();
    private int antallOrdTotalt = 0;
    public void lesFraFil (String filnavn) {
        In innfil = new In(filnavn);
        while (!innfil.lastItem()) {
            String ord = innfil.innWord();
            antallOrdTotalt++;
            if (alleOrd.containsKey(ord)) {
                Forekomster fo = (Forekomster) alleOrd.get(ord);
                fo.nyForekomst();
            } else {
                Forekomster fo = new Forekomster();
                alleOrd.put(ord, fo);
            }
        }
    }
    public void skrivTilFil (String filnavn) {
        Out utfil = new Out(filnavn);
        utfil.outln("Antall ord totalt: " + antallOrdTotalt);
        utfil.outln("Antall ulike ord: " + alleOrd.size());
        Iterator it = alleOrd.keySet().iterator();
        while (it.hasNext()) {
            String ord = (String) it.next();
            Forekomster fo = (Forekomster) alleOrd.get(ord);
            int antall = fo.fåAntall();
            if (antall > 5) {
                utfil.outln(ord + "\t" + antall);
            }
        }
        utfil.close();
    }
}

```

Merk at det i løsningsforslaget mangler typeparametre til HashMap og Iterator, siden det ble skrevet før typeparametre ble innført i Java. Riktige typeparametre er:  
 - HashMap<String,Forekomster>  
 - Iterator<String>