

## Inf1000, uke 5 – 19 sept. 2006

Innlevering av obligatoriske oppgaver - Joly  
Filbehandling  
Sudoku-eksempel med to-dimensjonal array  
Tekster

Arne Maus  
Institutt for informatikk  
Universitetet i Oslo

1

## Fordeler med elektronisk innlevering - Joly

- For studentene
  - Du får en kvittering på når du har levert hver oblig
  - Har du en slik kvittering, så kan ingen si at du ikke har levert
  - Obligen din blir lagret permanent i en database – går ikke tapt
- For INF1000, gruppelærerne, og Ifi
  - Vi får bedre orden på innlevering av obliger (sikring av innlevingene)
  - Vi får sjekket om noen innleverer kopier av andres besvarelser (uten Joly gjør minst 8-10% det)
  - Antall kopi-besvarelser synker sterkt, studentene gjør mer selv

Joly er skrevet av 4 master-studenter:

- Chr. K.Kielland, Hanne Vibekk, Theresese Stensen og Cato Morholt
- (slike systemer kan også kanskje du lage om 3 år)

2

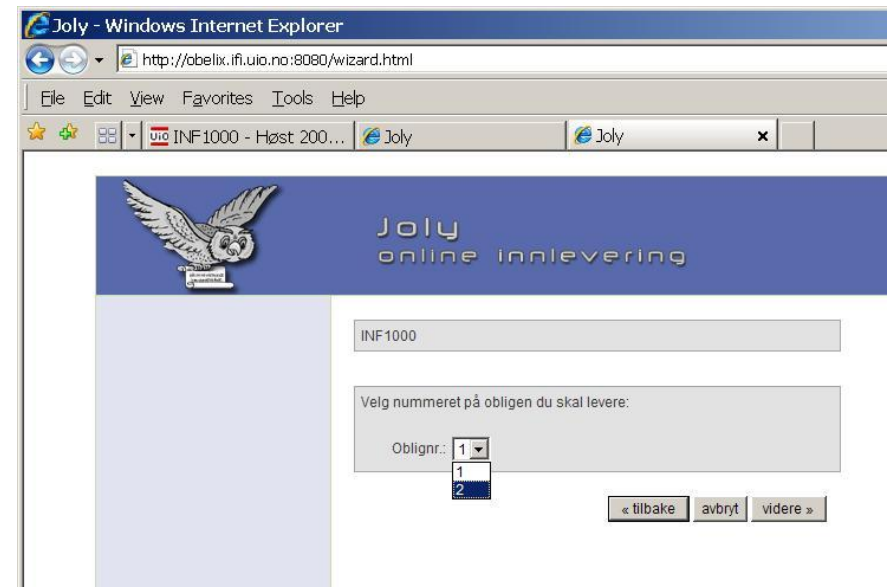
## Joly – innlevering av obligatoriske oppgaver <http://obelix.ifi.uio.no:8080/>

1) Velg kurs (h2006: bare INF1000)



3

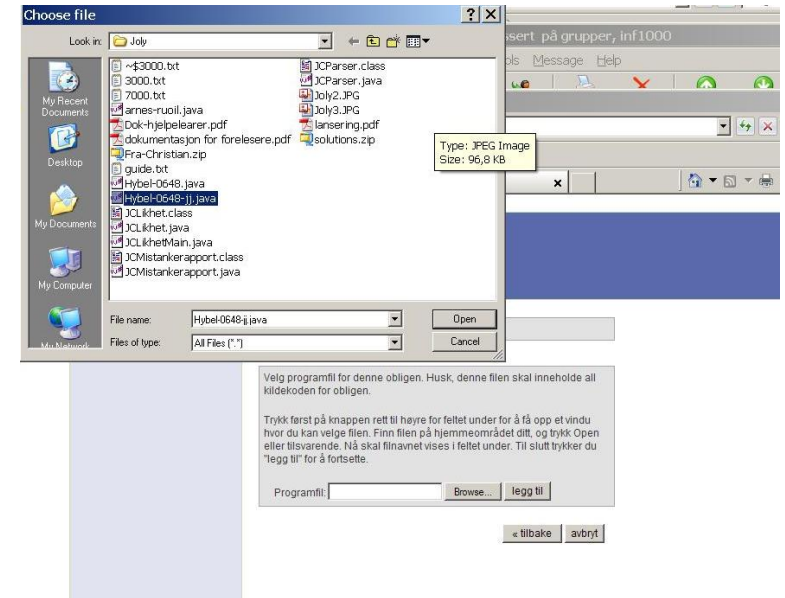
2) Velg gruppe



### 3a) Velg fil til oblig (hele stien med filnavnet sist)



### 3b) Velg fil til oblig (Trykk 'Browse' og let etter filen)



### 3c) Velg fil til oblig (Trykk 'Open' og så 'legg til')



### 4) Velg evt. tilleggs-fil til oblig (ikke vanlig, trykk bare videre)



### 5) Skriv brukernavn og evt. beskjed til gruppelærer.

INF1000 2 Hybel-0648-ij.java

Fyll ut brukernavnet ditt. Hvis du har kommentarer om innleveringen til gruppelæreren, kan du skrive disse i feltet under.

Brukernavn: kari

Beskjed til gruppelærer (valgfri): Min beste besvarelse var snill

< tilbake avbryt videre >

### 6) Sjekk opplysninger, trykk 'send inn'.

Registrerte opplysninger

Du har registrert følgende opplysninger. Vennligst kontroller at disse stemmer, og trykk "send inn" for å registrere obligen din.

Brukernavn:	kari
Gruppenr.:	1
Kurs:	INF1000
Oblignr.:	2
Programfil:	Hybel-0648-ij.java
Andre filer:	
Beskjed til gruppelærer (valgfri):	Min beste besvarelse var snill

< tilbake avbryt send inn

### 7) Kvittering, skriv ut og avslutt.

Kvittering for innlevert obligatorisk oppgave

Du har nå innlevert obligen. Du kan nå velge å skrive ut kvitteringen. Trykk da etterpå "Avslutt" for å komme tilbake til INF1000 hjemmesiden.

Kurs: INF1000

Oblignr.: 2

Programfil: Hybel-0648-ij.java

Andre filer:

Brukernavn: kari

Gruppenr.: 1

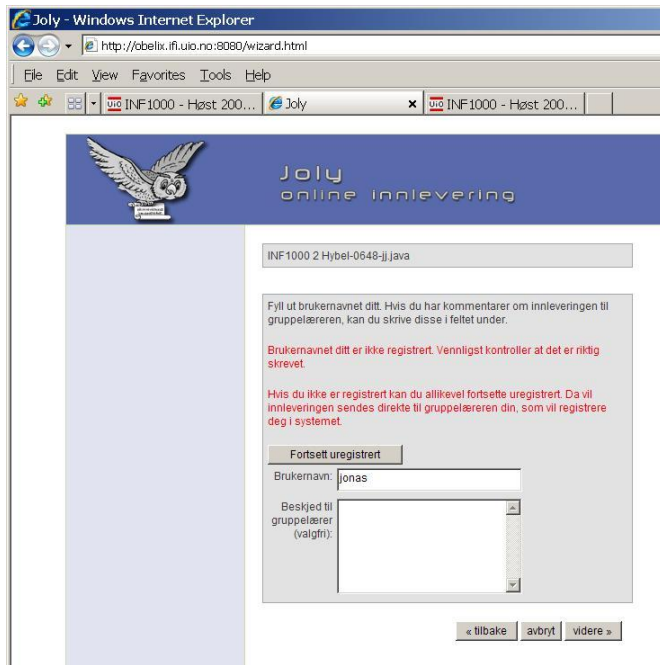
Beskjed til gruppelærer (valgfri): Min beste besvarelse var snill

Innleveringstidspunkt: Thu Sep 14 14:16:18 CEST 2006

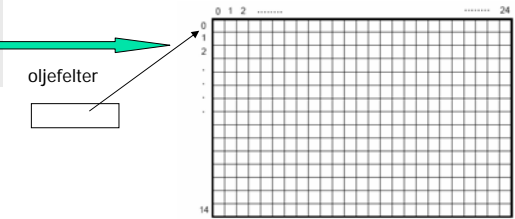
skriv ut Avslutt

## Hva hvis ??

- Jeg skal sende inn forbedret oblig
  - OK – alle blir lagret (og gruppelærer retter selvsagt siste)
- Hvordan legger jeg inn gruppenummer
  - Systemet finner selv hvilken gruppe du er påmeldt (og skal sende obliger til)
- Joly-systemet er nede når jeg skal levere
  - Levér på vanlig e-post til din gruppelærer (så legger han/hun) inn obligen din
- Hvis jeg ikke er registrert som student på gruppa
  - Da kan du fortsatt levere, men besvarelsen blir ikke lagret i basen
  - se neste foil
- Kan jeg levere hjemmefra
  - Sannsynligvis hvis du har VPN
  - (interessant spørsmål)

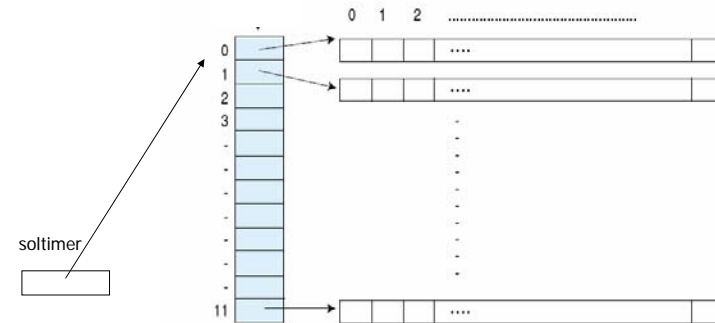


To-dimensjonale (2D) arrayer  
- slik tenker vi oss det  
- og slik er det



```
int[][] soltimer = new int[12][31];
```

gir følgende:



Figur 5.3 En todimensjonal array med arraypekere og arrayobjekter.

## Liste over ulike verdier i 2D-array

Vi skal lage et program som illustrerer hvordan man

- løper gjennom et 2D-array og ser på alle verdier
- legger inn i en (en-dimensjonal) array de ulike verdiene som forekommer i 2D-arrayen
- skriver ut listen med de ulike verdiene

Programmet skal:

- be om og lese inn verdier til 2D-arrayen
- skrive ut en sortert liste over de entydige verdier som forekommer i arrayen

## Programskisse

```
import easyIO.*;

class ArrayEksempel2 {
    public static void main (String[] args) {
        In tast = new In();

        int[][] a = new int[3][4];

        for (int i=0; i<3; i++) {
            for (int j=0; j<4; j++) {
                System.out.print("Gi et heltall: ");
                a[i][j] = tast.inInt();
            }
        }

        <lag liste over entydige verdier som forekommer
        i arrayen a, og skriv ut listen>

    }
}
```

## Å lage liste over entydige verdier

```
int[] verdier = new int[12];
int antall = 0;

for (int i=0; i<3; i++) {
    for (int j=0; j<4; j++) {
        int aij = a[i][j];
        boolean funnet = false;

        for (int k=0; k<antall; k++) {
            if (aij == verdier[k]) {
                funnet = true;
            }
        }

        if (!funnet) {
            verdier[antall] = aij;
            antall++;
        }
    }
}

for (int i=0; i<antall; i++) {
    System.out.println(verdier[i]);
}
```



17

## Lese – og skrive fra/til fil

- klassene In og Out i easyIO
- Les dokumentasjonen
  - In og Out + Format
    - brukes i INF1000
    - Format brukes til mer 'finjustert' formattering
  - InExp og OutExp
    - gir feilmeldinger hvis du gjør noen feil
    - vanskeligere å bruke enn In og Out
    - blir vanskeligere kode, brukes noe i INF1010
  - Det er langt flere metoder enn de som gjennomgås her
- easyIO ble laget fordi Javas innebygde IO-metoder var uhyre kompliserte
  - bedre nå, men enda noe vanskeligere enn easyIO

18

## Lese fra fil

```
import easyIO.*;

class LesFraFil {
    public static void main (String[] args) {
        In fil = new In("filnavn");

        String s = fil.inLine();
        System.out.println("Første linje var: " + s);
    }
}
```

Vi må først importere pakken easyIO

Vi åpner filen for lesing

Her leses hele første linje av filen



19

## Lese item for item

- Metoder:
  - inInt() for å lese et heltall
  - inDouble() for å lese et flyttall
  - inWord() for å lese et ord
  - inWord("\n") for å lese linje (hopper over alle tomme linjer)
  - lastItem() for å sjekke om slutten av filen er nådd
- Eksempel: lese en fil tall for tall

```
In fil = new In("fil.txt");
while (!fil.lastItem()) {
    int k = fil.inInt();
    System.out.println("Tallet var " + k);
}
```

20

## Eksempel

Program som leser en fil med to kolonner: en kolonne med desimaltall, og en kolonne med tekst.

```
import easyIO.*;

class LesFraFil2 {
    public static void main (String[] args) {
        In innfil = new In("fil.txt");
        double[] x = new double[100];
        String[] s = new String[100];

        int ant = 0;
        while (!innfil.lastItem()) {
            x[ant] = innfil.inDouble();
            s[ant] = innfil.inWord("\n");
            ant = ant + 1;
        }

        for (int i=0; i<ant; i++) {
            System.out.println(x[i] + s[i]);
        }
    }
}
```

21

## Lese linje for linje

- Metoder:
  - readLine() for å lese en linje
  - inLine() for å lese resten av en linje (leser neste linje hvis det ikke er mer igjen enn linjeskift på nåværende linje)
  - endOfFile() for å sjekke om slutten av filen er nådd
- Eksempel: lese en fil linjevis

```
In fil = new In("fil.txt");
while (!fil.endOfFile()) {
    String s = fil.readLine();
    System.out.println("Linjen var " + s);
}
```

22

## Eksempel

Program som leser en tekstfil linje for linje:

```
import easyIO.*;

class LesFraFil3 {
    public static void main (String[] args) {
        In innfil = new In("fil.txt");
        String[] s = new String[100];

        int ant = 0;
        while (!innfil.endOfFile()) {
            s[ant] = innfil.readLine();
            ant = ant + 1;
        }

        for (int i=0; i<ant; i++) {
            System.out.println(s[i]);
        }
    }
}
```

23

## Lese tegn for tegn

- Metoder:
  - inChar() for å lese et tegn (også blanke og linjeskift)
  - endOfFile() for å sjekke om slutten av filen er nådd
- Eksempel: lese en fil tegn for tegn

```
In fil = new In("fil.txt");
while (!fil.endOfFile()) {
    char c = fil.inChar();
    System.out.println("Tegnet var " + c);
}
```

24

## Eksempel

Program som leser en tekstfil tegn for tegn og skriver ut på skjerm, sammen med antall tegn i filen:

```
import easyIO.*;

class LesFraFil4 {
    public static void main (String [] args) {
        In innfil = new In("fil.txt");
        int antall = 0;

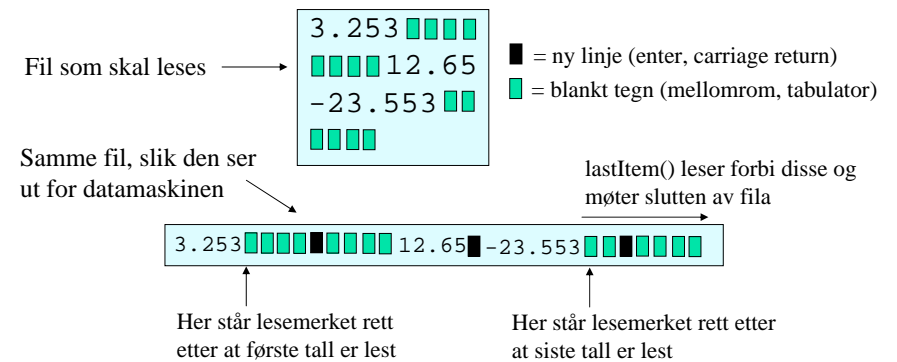
        while (!innfil.endOfFile()) {
            System.out.print(innfil.inChar());
            antall++;
        }

        System.out.println("\nAntall tegn: " + antall);
    }
}
```

25

## lastItem og endOfFile

- `endOfFile()` sjekker om siste tegn på fila er lest
- `lastItem()` søker seg fram til første ikke-blanke tegn og returnerer **true** hvis slutten av fila nås og **false** ellers.
- Eksempel:



26

## Når filens lengde er kjent

- Når et program skal lese en fil, må det ha en mulighet til å avgjøre når slutten av filen nådd - ellers kan det oppstå en feilsituasjon.
- Metodene `lastItem()` og `endOfFile()` kan benyttes til dette.
- Noen ganger er filens lengde kjent på forhånd:
  - lengden er kjent før programmet kjøres
  - lengden ligger lagret i begynnelsen av filen

Da kan vi i stedet benytte en for-løkke.

27

## Eksempel: fil med kjent lengde

Program som leser en fil med 10 desimaltall, hvor tallene er atskilt med blanke tegn og/eller linjeskift:

```
import easyIO.*;

class LesFraFil5 {
    public static void main (String[] args) {
        double[] x = new double[10];
        In innfil = new In("fil.txt");

        for (int i=0; i<10; i++) {
            x[i] = innfil.inDouble();
        }

        // Nå kan vi evt. gjøre noe med verdiene i arrayen x
    }
}
```

28



## Nok at tallene er atskilt

Programmet på forrige foil ville gitt akkurat samme resultat for alle disse filene:

15.2	15.2	6.23	15.2	6.23	3.522	3.6
6.23	3.522	3.6	8.893	-3.533	65.23	22.01
3.522	8.893	-3.533	65.23	22.01	45.02	7.2
3.6	65.23	22.01				
8.893	45.02	7.2				
-3.533						
65.23						
22.01						
45.02						
7.2						

29

## Eksempel: fil med lengde-info

Program som leser en fil med desimaltall, hvor tallene er atskilt med blanke tegn og/eller linjeskift. Antall tall som skal leses ligger øverst i filen.

```
import easyIO.*;

class LesFraFil6 {
    public static void main (String[] args) {
        double[] x; // bestemmer ikke lengden ennå
        In innfil = new In("fil.txt");

        int lengde = innfil.inInt(); // nå vet vi lengden
        x = new double[lengde];

        for (int i=0; i<lengde; i++) {
            x[i] = innfil.inDouble();
        }

        // Nå kan vi evt. gjøre noe med verdiene i arrayen x
    }
}
```

30

## Eksempel: fil med sluttmerke

Program som leser en fil med desimaltall, hvor tallene er atskilt med blanke tegn og/eller linjeskift. Slutten av filen er markert med tallet -999.

```
import easyIO.*;

class LesFraFil7 {
    public static void main (String [] args) {
        double [] x = new double[100]; // antar max 100 tall på fil
        In innfil = new In("fil.txt");
        double siste = 0;
        int ant = 0;

        while (siste != -999) {
            siste = innfil.inDouble();
            if (siste != -999) {
                x[ant] = siste;
                ant = ant + 1;
            }
        }
        // Nå ligger det verdier i x[0], x[1], ....., x[ant-1]
    }
}
```

31

## Lese en fil med mer komplisert format

- Anta at vi skal lese en fil med følgende format:
  - Først er det en linje med 3 overskrifter (separert av blanke tegn)
  - Deretter kommer det en eller flere linjer, som hver består av et heltall, et desimaltall og en tekststreng (separert av blanke tegn)

- Eksempel:

Antall	Pris	Varenavn
35	23.50	Oppvaskkost
53	33.00	Kaffe
97	27.50	Pizza
...	...	...
...	...	...

- Dataene som leses skal programmet ta vare på for senere formål.

32



## Framgangsmåte

- Den første linja er spesiell, og vi tenker oss her at den ikke er så interessant - vi ønsker bare å få lest forbi den. Det kan vi gjøre med `inLine()`.
- De andre linjene har samme format, så vi kan lage en løkke hvor hvert gjennomløp av løkken leser de tre itemene på en linje. Vi bruker da henholdsvis `inInt()`, `inDouble()` og `inWord()`.
- For å vite når filen er slutt, kan vi enten bruke `endOfFile()` eller `lastItem()`. Siden vi leser filen itemvis, er det mest naturlig å bruke `lastItem()`. Da får vi heller ikke problemer dersom det skulle ligge noen blanke helt på slutten av filen.
- Vi hopper over detaljene.

33

## Eksempel på gi skilletegn ved innlesing

- Vi kan ved innlesing i `easyIO` spesifisere hvilke tegn vi vil hoppe over ved innlesing – i `inInt`, `inWord`, `inDouble`,... kan skilletegn gis
- Anta at kolumnen `k` og raden `r` skal gis som `S(r,k)` – eks: `S(0,4)`

```
import easyIO.*;

class Skilletegn {
    public static void main (String[] args) {
        int r,k;
        String skille = " \nS(,)";
        In tast = new In(); Out skjerm = new Out();

        skjerm.out("Gi rad r og kolumnen k som S(r,k):");
        r = tast.inInt(skille);
        k = tast.inInt(skille);
        skjerm.out("Du ga r=" +r+", og k=" +k);
    }
}
```

4

## Noen nyttige hjelpemidler (ikke pensum)

- Sjekk om det finnes en fil med et bestemt navn:

```
if (new File("filnavn").exists()) {
    System.out.println("Filen finnes");
}
```

- Slette en fil:

```
if (new File("filnavn").delete()) {
    System.out.println("Filen ble slettet");
}
```

- Avgjøre hvilket filområde programmet ble startet fra:

```
String curDir = System.getProperty("user.dir");
```

- Lage liste over alle filer og kataloger på et filområde:

```
String [] allefiler = new File("filområdenavn").list();
```

Merk: klassen `File` ligger i pakken `java.io` som derfor må importeres først.

35

## Skrive til fil

```
import easyIO.*;

class SkrivTilFil {
    public static void main (String [] args) {
        Out fil = new Out("filnavn");
        fil.outln("Dette er første linje");
        fil.close();
    }
}
```

Vi må først importere pakken `easyIO`

Vi åpner filen for skriving

Her skrives en linje med tekst til filen

Vi må huske å lukke filen til slutt

36

## Hvilke skrivemetoder finnes?

Data-type	Eksempel	Beskrivelse
int	fil.out(x); fil.out(x, 6);	Skriv x Skriv x høyrejustert på 6 plasser
double	fil.out(x, 2); fil.out(x, 2, 6);	Skriv x med 2 desimaler Skriv x med 2 desimaler på 6 plasser
char	fil.out(c);	Skriv c
String	fil.out(s); fil.out(s, 6);	Skriv s Skriv s på 6 plasser (venstrejustert)
	fil.outln();	Skriv en linjeskift
	fil.close();	Lukk filen

**Merk:** dersom antall plasser spesifiseres og det ikke er plass til det som skal skrives ut, vil det som skrives ut avsluttes med tre punktumer: ...

37

## Tekster og klassen String

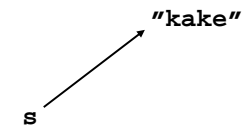
- En tekststreng er en sekvens av tegn (null, en eller flere), f.eks.

""

"&"

"Kaia er student"

- Hver tekststreng vi lager er et *objekt* av typen String
- En String-variabel (f.eks. `string s`) er en *referanse* til et slikt objekt
- Resultatet av å utføre `String s = "kake"` :



- For å finne lengden (dvs antall tegn i) en tekst:

```
int lengde = s.length();
```

38

## Bruk av spesialtegn

- Både i char-uttrykk og String-uttrykk kan vi ha mange ulike typer tegn
- Alle Unicode-tegn er tillatt
- Unicode er en standard som tillater tusenvis av tegn (ulike varianter fins; den som støttes av Java tillater 65536 ulike tegn)
- Alle tegnene kan angis som `'\uxxxx'` hvor hver x er en av 0, 1, 2, ..., 9, A, B, C, D, E, F  
Eksempel: `'\u0041'` er tegnet 'A'
- Noen spesialtegn har egen forkortelse:
  - `\t` tabulator
  - `\r` vognretur (skrivning starter først på linja)
  - `\n` linjeskift
  - `\"` dobbelt anførselstegn
  - `\'` enkelt anførselstegn
  - `\\` bakslask

39

## Unicode (<http://www.unicode.org>)

Three tables showing Unicode character sets. The first table shows basic Latin characters (0-127) with some highlighted in red. The second table shows Latin-1 Supplement (128-255) with some highlighted in green. The third table shows Latin Extended-A (256-383).

40

