

Inf1000 (Uke 5)

Arrayer, filer og tekst

Innlevering av obligatoriske oppgaver – Joly

Arrayer

Filbehandling

Tekster

Are Magnus Bruaset og Arild Waaler

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

Velkommen til Joly - online innlevering

Denne assistenten vil ta deg gjennom de nødvendige stegene for å levere en oblig. Først må du velge hvilket kurs du skal levere oblig i:

Kurs: INF1000

videre »

3

2) Velg gruppe

INF1000

Velg nummeret på obligen du skal levere:

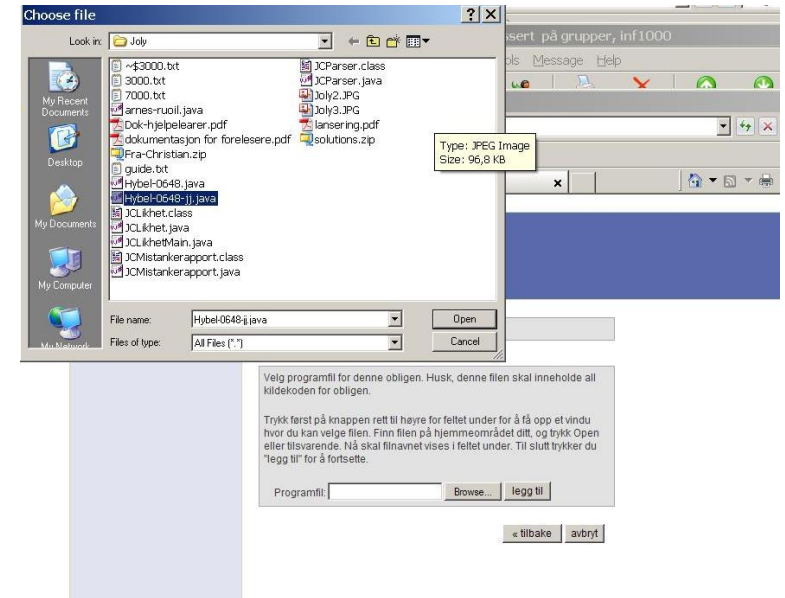
Oblignr.: 1

« tilbake avbryt videre »

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



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



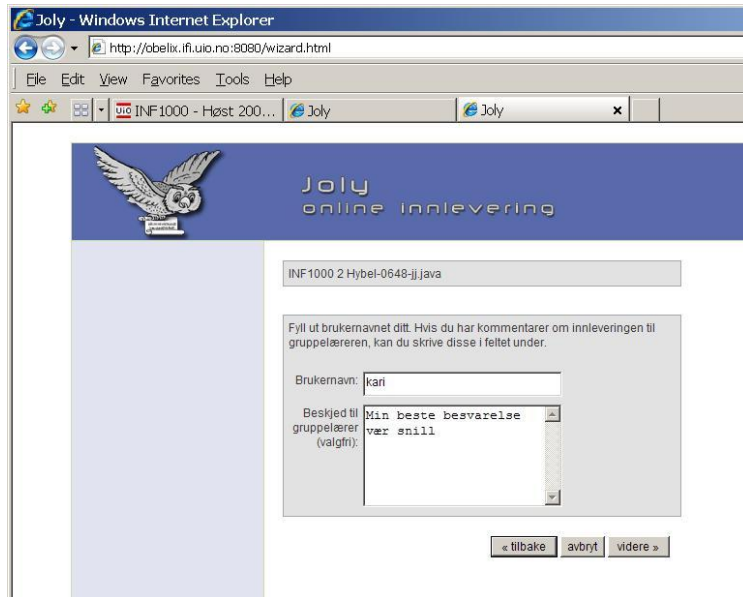
3c) Velg fil til oblig (Trykk "Open' og så 'legg til')



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



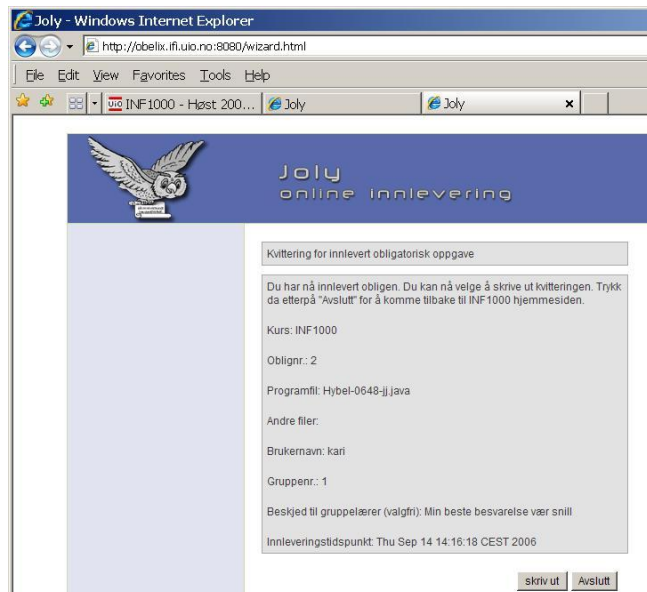
5) Skriv brukernavn og evt. beskjed til gruppelærer.



6) Sjekk opplysninger, trykk 'send inn'.

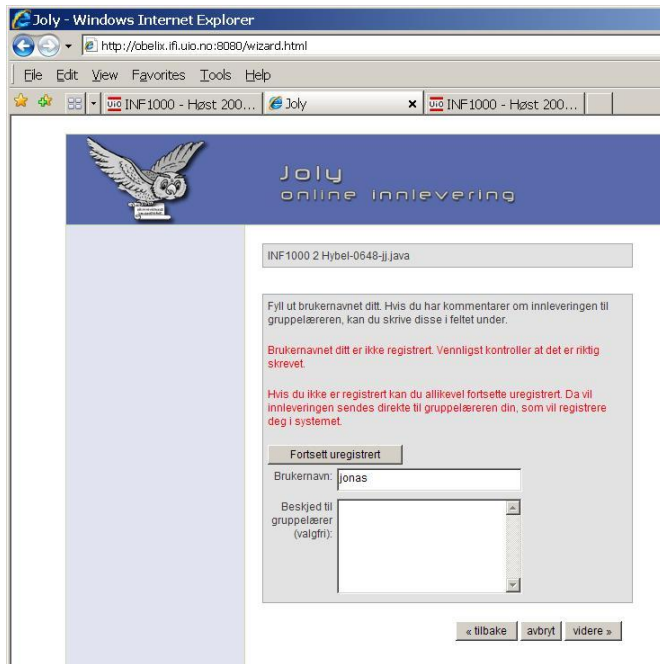


7) Kvittering, skriv ut og avslutt.



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)



Arrayer

- Hittil har vi sett på variable som kan holde en enkelt verdi:
 - en int-variabel har plass til ett heltall
 - en String-variabel har plass til en enkelt tekststreng
 - osv.
- Arrayer er "variable" som kan holde på mange verdier:
 - en int-array har plass til mange heltall
 - en String-array har plass til mange tekststrenger
 - osv.
- Verdiene som ligger i en array har hver sin posisjon (= indeks):
 - 0, 1, 2,, K-1 hvor K = lengden til arrayen
- En array x med lengde 4 kan visualiseres slik:



14

Deklarere og opprette arrayer

- Deklarere en array (gi den et navn):

```
datatype [] arrayNavn;
```
- Opprette en array (sette av plass i hukommelsen):

```
arrayNavn = new datatype [K]; // K er ønsket lengde
```
- Deklarere og opprette i en operasjon:

```
datatype [] arrayNavn = new datatype [K];
```
- Eksempler:


```
int [] a = new int [10];
double [] x = new double [100];
String [] s = new String [1000];
```

15

Verdiene i en array

- Anta at vi har deklartert og opprettet følgende array:


```
int [] tlf = new int [600];
```
- For å få tak i de enkelte verdiene i arrayen:


```
tlf [0], tlf [1], tlf [2], ..., tlf [599]
```
- For å få tak i lengden på arrayen:


```
tlf.length // NB: ingen parenteser til slutt
```
- For å sortere elementene i en array (i stigende rekkefølge):


```
java.util.Arrays.sort (tlf);
```

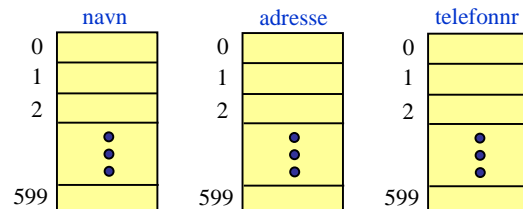
16

Eksempel på bruk av arrayer

- Anta at vi ønsker å lagre navn, adresse og telefonnr for de som følger et bestemt kurs med maksimalt 600 studenter:

```
String[] navn = new String[600];
String[] adresse = new String[600];
int[] telefonnr = new int[600];
```

- Resultatet kan visualiseres slik:



17

Automatisk initialisering av arrayer

- Når en array blir opprettet, blir den automatisk initialisert (dvs verdiene er ikke udefinerte når den er opprettet).

```
int[] k = new int[100]; // Nå er alle k[i] == 0
double[] x = new double[100]; // Nå er alle x[i] == 0.0
boolean[] b = new boolean[100]; // Nå er alle b[i] == false
char[] c = new char[100]; // Nå er alle c[i] == '\u0000'
String[] s = new String[100]; // Nå er alle s[i] == null
```

- Merk: String-arrayer initialiseres med den spesielle verdien `null`. Dette er *ikke* en tekststreng og må ikke blandes sammen med en tom tekst: `""`.
- For å kunne bruke verdien `s[i]` til noe fornuftig må du først sørge for å gi `s[i]` en tekststreng-verdi, f.eks. `s[i] = "Per"` eller `s[i] = ""`.

18

Egendefinert initialisering av en array

- Det er ikke alltid den automatiske initialiseringen av en array gir det vi ønsker. Vi kan da initialisere arrayen med våre egne verdier, slik som i disse eksemplene:

```
int[] printall = {2, 3, 5, 7, 11, 13};

double[] halve = {0.0, 0.5, 1.0, 1.5, 2.0};

String[] ukedager = {"Mandag", "Tirsdag", "Onsdag",
                    "Torsdag", "Fredag", "Lørdag",
                    "Søndag"};
```

19

Eksempel: lese og skrive ut

- Program som leser fem navn fra terminal og skriver dem ut igjen:

```
import easyIO.*;

class LesOgSkriv {
    public static void main (String [] args) {
        In tastatur = new In();
        String[] s = new String[5];

        for (int i=0; i<5; i++) {
            System.out.print("Navn: ");
            s[i] = tastatur.inLine();
        }

        for (int i=0; i<5; i++) {
            System.out.println(s[i]);
        }
    }
}
```

20

Eksempel: lese og skrive ut sortert

- Program som leser fem navn fra terminal og skriver dem ut igjen:

```
import easyIO.*;

class LesOgSkriv {
    public static void main (String [] args) {
        In tastatur = new In();
        String[] s = new String[5];

        for (int i=0; i<5; i++) {
            System.out.print("Navn: ");
            s[i] = tastatur.inLine();
        }

        java.util.Arrays.sort(s);
        for (int i=0; i<5; i++) {
            System.out.println(s[i]);
        }
    }
}
```

21

Eksempel: finne en bestemt verdi

```
/* a er en int-array og x er en int-variabel. Vi ønsker å sjekke om verdien i x forekommer i a
*/

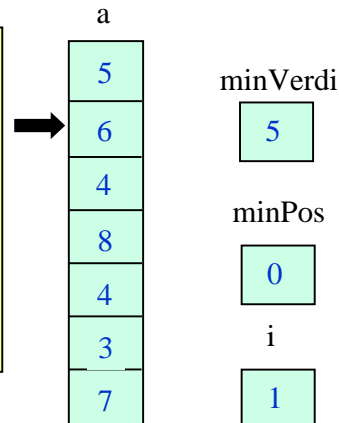
boolean funnet = false;
int i = 0;
while (i<a.length && !funnet) {
    if (a[i] == x) {
        funnet = true;
    }
    i++;
}
if (funnet) {
    System.out.println("Verdien ligger i posisjon " + (i-1));
} else {
    System.out.println("Verdien ble ikke funnet!");
}
```

Eksempel: finne den minste verdien

- Følgende eksempel illustrerer hvordan man kan finne den minste verdien i en array:

```
//Anta at x er en double-array
double minVerdi = a[0];
int minPos = 0;
for (int i=1; i<a.length; i++) {
    if (a[i] < minVerdi) {
        minVerdi = a[i];
        minPos = i;
    }
}

// Nå er minVerdi lik den minste verdien i x
// og minPos er posisjonen den ligger i
```



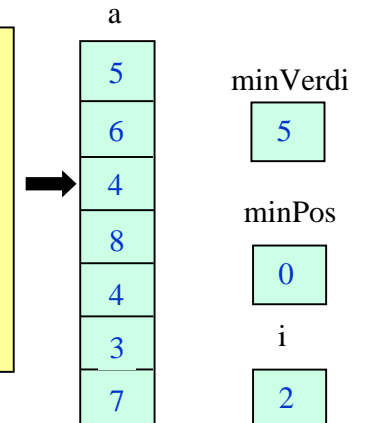
23

Eksempel: finne den minste verdien

- Følgende eksempel illustrerer hvordan man kan finne den minste verdien i en array:

```
//Anta at x er en double-array
double minVerdi = a[0];
int minPos = 0;
for (int i=1; i<a.length; i++) {
    if (a[i] < minVerdi) {
        minVerdi = a[i];
        minPos = i;
    }
}

// Nå er minVerdi lik den minste verdien i x
// og minPos er posisjonen den ligger i
```

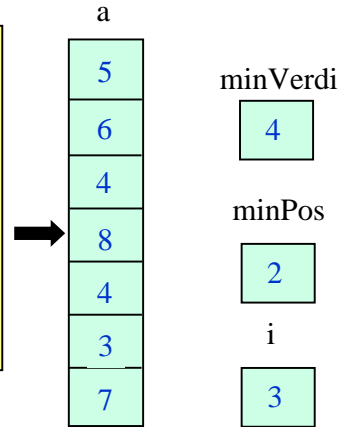


24

Eksempel: finne den minste verdien

- Følgende eksempel illustrerer hvordan man kan finne den minste verdien i en array:

```
//Anta at x er en double-array
double minVerdi = a[0];
int minPos = 0;
for (int i=1; i<a.length; i++) {
    if (a[i] < minVerdi) {
        minVerdi = a[i];
        minPos = i;
    }
}
// Nå er minVerdi lik den minste verdien i x
// og minPos er posisjonen den ligger i
```

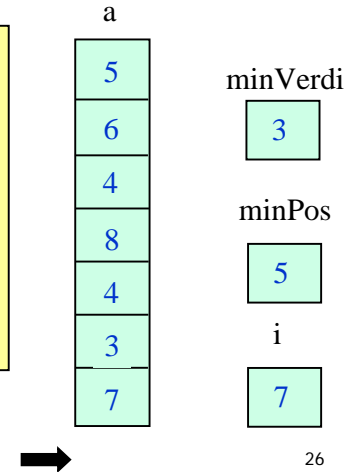


25

Eksempel: finne den minste verdien

- Følgende eksempel illustrerer hvordan man kan finne den minste verdien i en array:

```
//Anta at x er en double-array
double minVerdi = a[0];
int minPos = 0;
for (int i=1; i<a.length; i++) {
    if (a[i] < minVerdi) {
        minVerdi = a[i];
        minPos = i;
    }
}
// Nå er minVerdi lik den minste verdien i x
// og minPos er posisjonen den ligger i
```



26

Eksempel: ferdig program

```
import easyIO.*;
class FinnMinsteVerdi {
    public static void main (String [] args) {
        In tastatur = new In();
        double[] a = new double[5];

        for (int i=0; i<a.length; i++) {
            System.out.print("Oppgi en verdi: ");
            a[i] = tastatur.inDouble();
        }

        double minVerdi = a[0];
        int minPos = 0;
        for (int i=1; i<a.length; i++) {
            if (a[i] < minVerdi) {
                minVerdi = a[i];
                minPos = i;
            }
        }
        System.out.println("Minste verdi er " + minVerdi);
        System.out.println("Den ligger på plass " + minPos);
    }
}
```

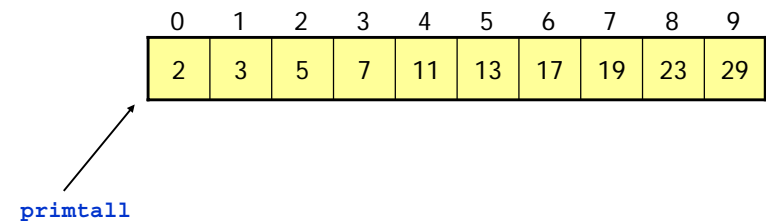
27

En array-variabel er en adresse

- Når vi deklarerer en array så refererer arraynavnet ikke til selve verdiene i arrayen, men til adressen (i hukommelsen) hvor verdiene ligger lagret.
- Resultatet etter at vi har utført

```
int[] printtall = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29};
```

kan visualiseres slik:



28

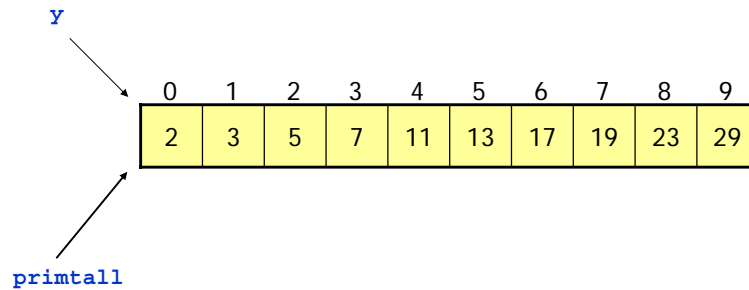
Å kopiere en array-adresse

- Setningen

```
int [] y = printall;
```

lager et nytt *navn* på arrayen.

Den lager en kopi av adressen, ikke innholdet:



29

Oppgave

Hva blir utskriften fra følgende program?

```
class ToArrayer {  
    public static void main (String [] args) {  
        int[] x = new int[10];  
        int[] y = x;  
        for (int i=0; i<10; i++) {  
            x[i] = i;  
        }  
        for (int i=0; i<10; i++) {  
            System.out.println(y[i]);  
        }  
    }  
}
```

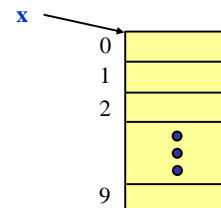
30

Hva som skjedde

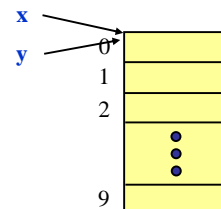
Etter å ha utført instruksjonen

.... så er situasjonen denne:

```
int[] x = new int[10];
```



```
int[] y = x;
```



31

Kompilering og kjøring

```
> javac ToArrayer.java  
> java ToArrayer  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

32

Kopiering av arrayer

Vi kan ikke lage en kopi av en array x ved å skrive

```
int[] y = x;
```

siden dette bare medfører at adressen til arrayen legges inn i y.

- Skal vi lage en kopi, må vi først opprette en array til (f.eks. y), og så kopiere over verdiene en for en:

```
double[] y = new double[x.length];
for (int i=0; i<x.length; i++) {
    y[i] = x[i];
}
```

- Det finnes også ferdige verktøy i Java for å kopiere en array, f.eks:

```
int[] y = (int[]) x.clone();
```

33

Når arrayen blir for liten

- Noen ganger får vi behov for å utvide en array.
- Framgangsmåte for å utvide en array:
 - Deklarer og opprett en ny array **temp** som er av ønsket lengde
 - Flytt over alle verdier fra opprinnelig array til **temp**
 - Sett opprinnelig array-peker til å peke på **temp**
- Programkode:

```
/* Anta at tall er en int-array og at vi ønsker
   å utvide tall til dobbel lengde */

int[] temp = new int[2 * tall.length];
for (int i=0; i<tall.length; i++) {
    temp[i] = tall[i];
}
tall = temp;
```

34

Flerdimensjonale arrayer

- Vi kan også deklarere todimensjonale (og høyeredimensjonale) arrayer.
- Eksempel:

```
String[][] eier = new String[15][25];
```

- Resultat:

eier

	0	1	2	3	4	...	24
0						
1						
2						
3						
4						
5						
6						
7						
.						
.						
.						
14						

Hvis et felt ikke er solgt:
`eier[i][j] == null`

Hvis et felt er solgt:
`eier[i][j] != null`

- Eksempler på lovlige operasjoner:

```
eier[3][4] = "Petrol A/S";
int antallRader = eier.length;
int antallKolonner = eier[0].length;
```

35

Eksempel: finn antall solgte felt

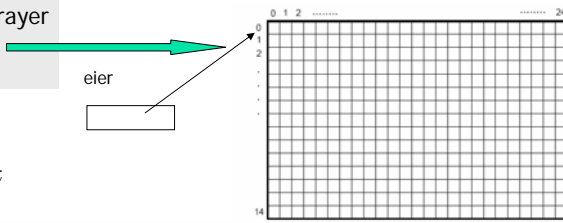
```
class AntallFelt {
    public static void main (String [] args) {
        String[][] eier = new String[15][25];
        <innlesning av eiere m.m.>

        int antallSolgte = 0;
        for (int i=0; i<15; i++) {
            for (int j=0; j<25; j++) {
                if (eier[i][j] != null) {
                    antallSolgte++;
                }
            }
        }
        System.out.println("Antall solgte felt: " + antallSolgte);
    }
}
```

36

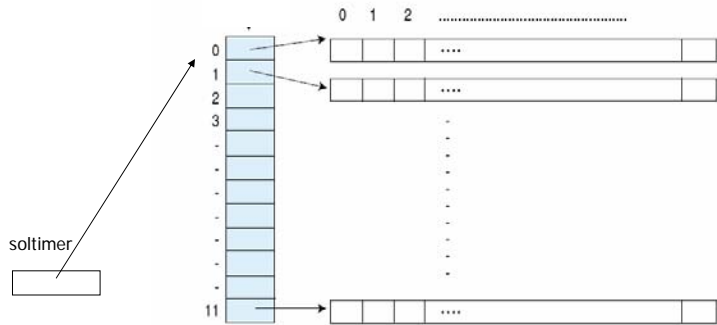
To-dimensjonale (2D) arrayer

- 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 liste over verdiene som forekommer i arrayen slik at hver verdi kun skrives ut en gang

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]);
}
```

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

41

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

42

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);
}
```

43

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]);
        }
    }
}
```

44

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);
}
```

45

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]);
        }
    }
}
```

46

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);
}
```

47

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;

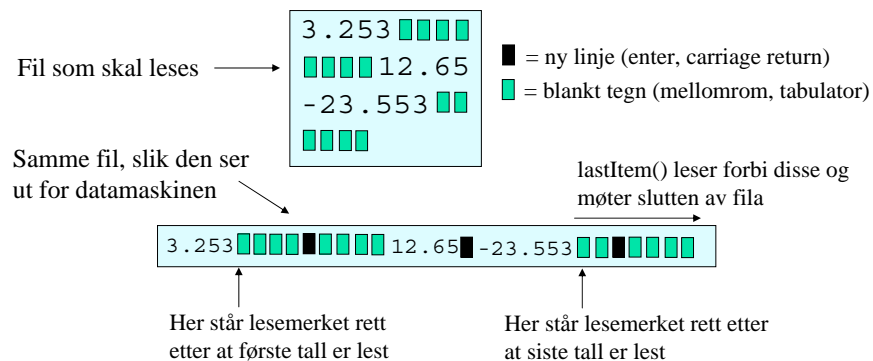
        while (!innfil.endOfFile()) {
            System.out.print(innfil.inChar());
            antall++;
        }

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

48

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:



49

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.

50

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
    }
}

```

51

Nok at tallene er atskilt

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

15.2			
6.23			
3.522			
3.6			
8.893			
-3.533			
65.23			
22.01			
45.02			
7.2			

15.2	6.23		
3.522	3.6		
8.893	-3.533		
65.23	22.01		
45.02	7.2		

15.2	6.23	3.522	3.6
8.893	-3.533		
65.23	22.01	45.02	
			7.2

52

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
    }
}
```

53

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]
    }
}
```

54

Lese en fil med mere 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.

55

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.

56

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

7

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.

58

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

Vi må huske å lukke filen til slutt

Her skrives en linje med tekst til filen

59

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); fil.outln();	Skriv s Skriv s på 6 plasser (venstrejustert) 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: ...

60

Tekster og klassen String

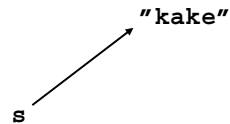
- En tekststreng er en sekvens av tegn (null, en eller flere), f.eks.

""

"&"

"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();
```

61

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

62

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

Three tables showing parts of the Unicode character set. The first table shows Latin characters from U+0000 to U+007F. The second table shows Latin characters with diacritics from U+0080 to U+00FF. The third table shows Latin characters with diacritics from U+0100 to U+017F. Some characters are highlighted with green boxes.

63

Andre eksempler på Unicode-tegn

Tibetanske tegn

A table showing Tibetan characters from U+0F00 to U+0FFF. The characters are arranged in a grid with column headers U+0F00 to U+0FFF and row headers 0 to F.

Musikksymboler

A table showing musical symbols from U+1D00 to U+1D7F. The characters are arranged in a grid with column headers U+1D00 to U+1D7F and row headers 0 to F.

64

Teste om to tekster er like

- For å teste om to tekststrenger er like, brukes equals:

```
// Anta at s og t er tekstvariable (og at s ikke har verdien null)
if (s.equals(t)) {
    System.out.println("Tekstene er like");
} else {
    System.out.println("Teksten er forskjellige");
}
```

- Bruk av == virker av og til, men ikke alltid:

```
String s = "abc";
String t = "def";
String tekst1 = s + t;
String tekst2 = s + t;
```

Nå er `tekst1.equals(tekst2)` true, mens `tekst1 == tekst2` er false.

65

De enkelte tegnene i en tekststreng

- Tegnene i en tekststreng har posisjoner indeksert fra 0 og oppover

0	1	2	3
'k'	'a'	'k'	'e'

- Vi kan få tak i tegnet i en bestemt posisjon:

```
String s = "kake";
char c = s.charAt(1);
// Nå er c == 'a'
```

- Vi kan erstatte alle forekomster av et tegn med et annet tegn:

```
String s1 = "kake";
String s2 = s1.replace('k', 'r');
// Nå er s2 en referanse til tekststrengen "rare"
```

66

Deler av en tekststreng

- Vi kan trekke ut en del av en tekststreng:

```
String s = "Paris";
String s1 = s.substring(1,4);
// Nå er s1 tekststrengen "ari"
```

0	1	2	3	4
'P'	'a'	'r'	'i'	's'

`s.substring(1,4)`

- Generelt:

```
s.substring(index1, index2)
```

Første posisjon som
skal være med

Første posisjon som
ikke skal være med

- Siste del av en tekststreng:

```
String s = "Paris er hovedstaden i Frankrike";
String s1 = s.substring(6);
// Nå er s1 tekststrengen "er hovedstaden i Frankrike"
```

67