



PRG1000B Grunnleggende programmering

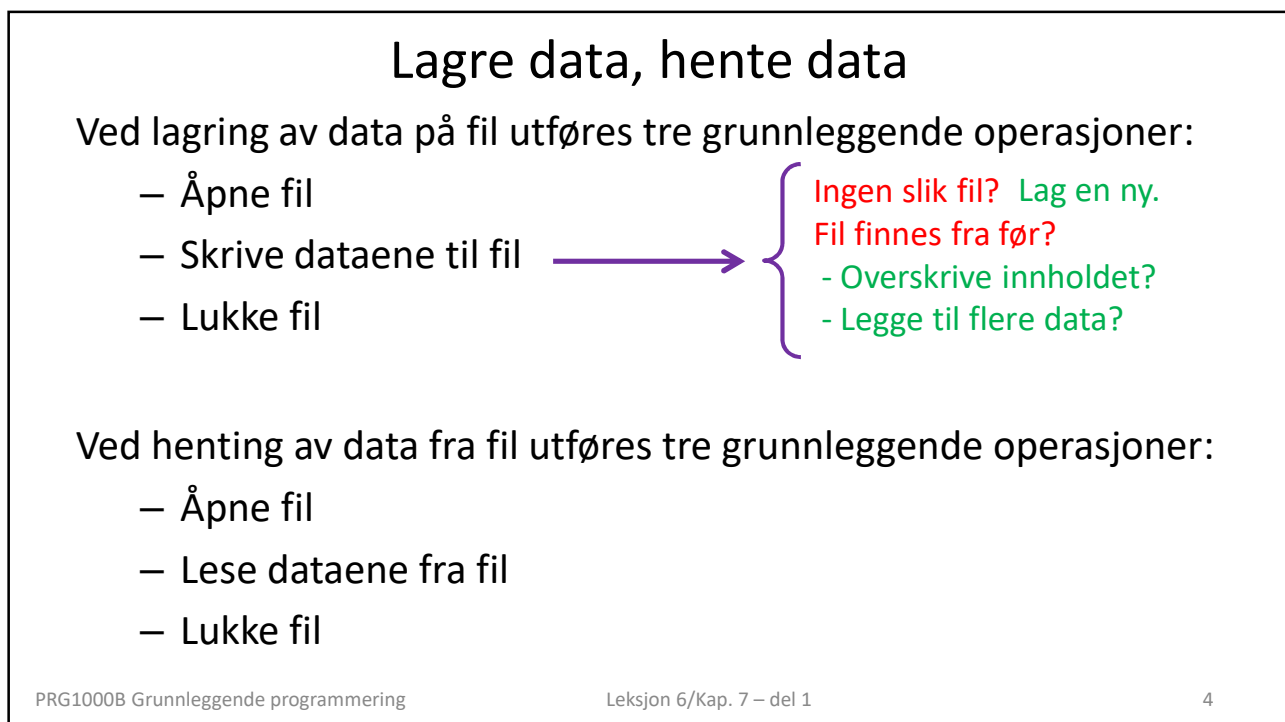
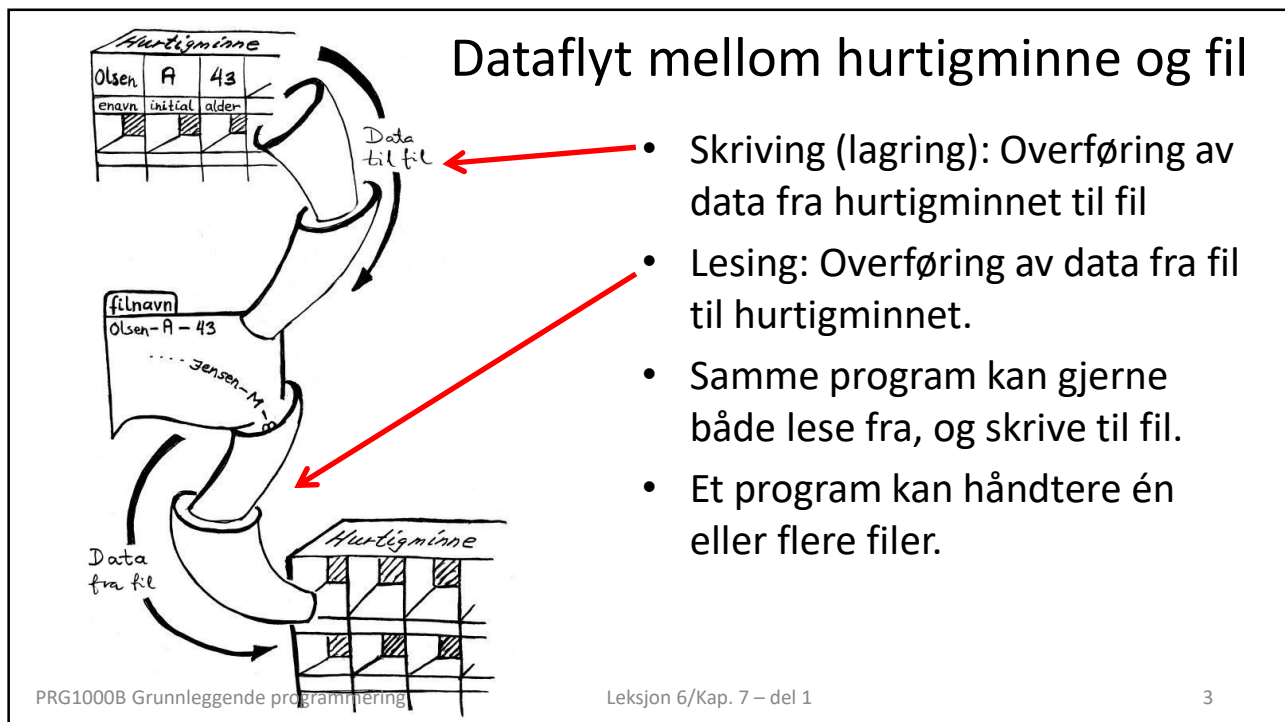
## Leksjon 6 / Kap. 7

# Filer

Roy M. Istad, 2017

### Fil – permanent lagring

- Ønsker at program skal kunne ta vare på data over tid, fra en kjøring til den neste (kontra hurtigminnet som har midlertidig lagring).
- Filer brukes for å lagre data permanent på ytre lager (f.eks. harddisk).
- Det kan være vanlige tekstfiler som er lesbare – for oss – i en vanlig tekstbehandler/editor.
- Data kan også lagres på binærfiler. Disse kan kun leses av spesielle program (slike filer sier man er lagret i et *proprietært format*).



## Eksempelkode: Lagre data

Skrive til fil

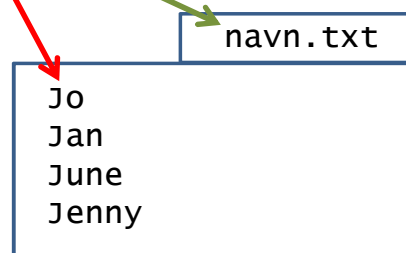
```
import java.io.*; // Importerer aktuelle filklasser

public class SkrivNavneliste {
    public static void main(String[] args) throws Exception {
        // 1) Åpner filen for skriving
        String filnavn = "navn.txt";
        PrintWriter skriver = new PrintWriter(filnavn);

        String[] navnTab = {"Jo", "Jan", "June", "Jenny"};

        // 2) Skriv hvert navn i navnTab til fil
        for (int i=0; i<navnTab.length; i++)
            skriver.println( navnTab[i] );

        // 3) Lukker filen
        skriver.close();
    }
}
```



## Unntak: Første møte

**Problem:** Opprette fil på en underkatalog som ikke finnes?

Forsøk på å lage PrintWriter-objektet feiler, og programutførelsen reagerer med å *kaste et unntak* (et objekt for en feilsituasjon).

I toppen av main-metoden er det føyd til en *throws-klausul*:

```
public static void main(String[] args) throws Exception
```

Dvs. main-metoden inneholder aktivitet som kan kaste unntak. Men, om et problem oppstår, skal unntaket bare kastes videre (throws).

Dette vil medføre at programkjøringen blir avbrutt med en feilmelding (*runtime error*).

Vi ser senere på hvordan unntak håndteres med en **try-catch**-setning.

## Usjekkede/sjekkede unntak

Program har «krasjet» før også.

F.eks. ved heltallsdivisjon med 0, eller tabellindekser som har vært utenfor lovlig område.

Hvorfor var ikke unntak tema der og da?

- **Usjekkede unntak:** Problem som *ikke må* håndteres i programkoden – medfører programkrasj om de oppstår.
- **Sjekkede unntak:** Problemsituasjoner som må håndteres i programkoden
  - throws-klausul (videreformidlet, ikke løst - førstevalget)
  - try-catch-setning (mer detaljert problemløsning – senere)

## Skrive til fil

```
String fil = "filnavn.txt";
```

1) Filen finnes ikke fra før: Ny fil blir opprettet.

```
PrintWriter skriver = new PrintWriter(fil);
```

2) Filen finnes fra før.

- Overskrive eksisterende innhold:

```
PrintWriter skriver = new PrintWriter(fil);
```

- Føye til mer innhold, henge på nye linjer:

```
FileWriter skrivelink = new FileWriter(fil, true);
PrintWriter skriver = new PrintWriter(skrivelink);
//Alt. new PrintWriter(new FileWriter(fil, true));
```

«append» -  
heng på

## Lese fra fil

tekstfil.txt

Dette er teksten som utgjør linje 1.  
Her er linje 2  
og dette er linje 3.  
Så kommer fjerde og siste linje.

Kvasikode:

Åpne fila for lesing

Les inn (første) linje fra fila

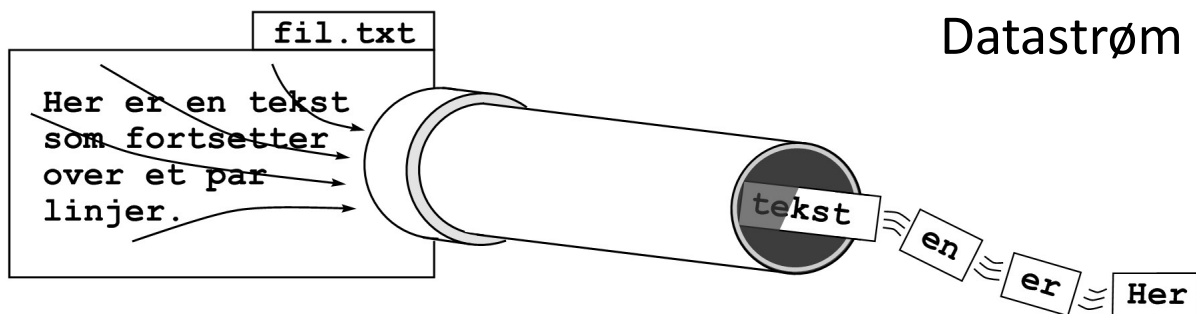
Gjenta til slutten på fila

Behandle innlest linje

Les inn (neste) linje fra fila

Lukk fila

Linjesluttmerke: \n



## Datastrøm

Ved lesing «strømmer» data fra fil til hurtigminnet

Ved skriving strømmer data fra hurtigminnet til fil.

Hva strømmer? Strøm av linjer, strøm av ord, eller bare en strøm av byter.

## Scanner-klassen

- Velegnet til lesing av tekstfiler (datastrøm fra fil)
  - Krever: `import java.util.*;`
- Opprett et Scanner-objekt kalt «leser»
  - `Scanner leser = new Scanner(aktuelleFil);`
  - `while ( leser.hasNext() )`
    - `true`: fortsett så lenge det er noe mer (String-data) på filen og bruk `leser.next()` til å hente neste tegnsekvens fra filen
    - `false`: avslutt data-henting fra filen
  - Alternativ: Linjevis håndtering av data fra fil (en hel linje om gangen)
    - Så lenge `leser.hasNextLine()` er `true`, bruk `leser.nextLine()` til å hente neste linje (String-data) fra filen
- Scanner-objekt kan også håndtere String som inndata!
- [Listing 7.2, side 274](#): VisLinjeNr

tekstfil.txt

```

Dette er på linje 1.
Her er linje 2,
og dette er linje 3.
Her er siste linje.

```

Linjesluttmerke: \n

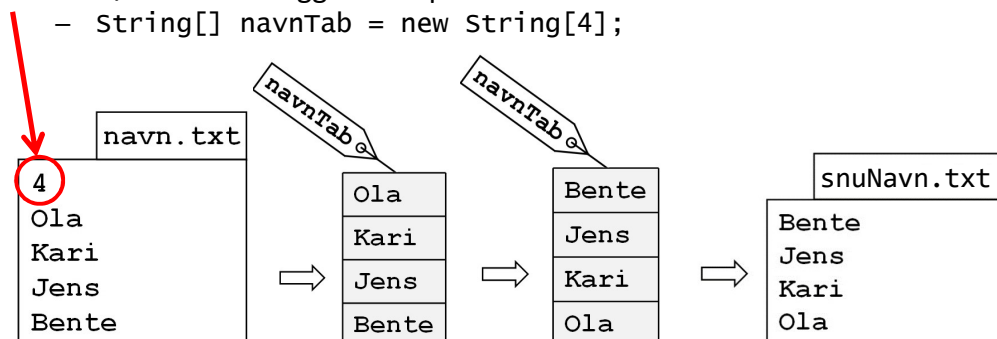
## Fra fil til datastruktur

Hvordan «snu» en navneliste lagret på fil?

- lese alle navnene inn i en tabell (i hurtigminnet)
- snu tabellen (motsatt alfabetisk rekkefølge)
- skrive alle navnene tilbake til filen igjen

Tabellstørrelse? Ligger først på filen ...

- `String[] navnTab = new String[4];`



## Snu navneliste «på fil».

```
File innFil = new File("navn.txt");
Scanner leser = new Scanner(innFil); // NB! Filinnpakning først

// Antall navn ligger på første linje
int antNavn = parseInt( leser.nextLine() );
String[] navnTab = new String[antNavn];

for (int pos=0; pos<antNavn; pos++)
    navnTab[pos] = leser.nextLine();
leser.close(); // Tabellen er fylt med navn

// Tabellen snus
PrintWriter skriver = new PrintWriter("snuNavn.txt");
for (int pos=antNavn-1; pos>=0; pos--)
    skriver.println(navnTab[pos]);
skriver.close();
```

navn.txt

4  
Ola  
Kari  
Jens  
Bente

snuNavn.txt

Bente  
Jens  
Kari  
Ola

PRG1000B Grunnleggende programmering

Leksjon 6/Kap. 7 – del 1

13

## Lese typede data

### Eks. Heltall på fil

tallfil.txt

23 40 17 82  
17  
20 47  
12 7 88 0 12 56

Splitter tekstlinje (String)  
parseInt(String)

→ sum = 0 + 23

- Alle data på filen oppfattes å ha samme datatype  
Spesielt: String, int, double

Default: Verdier adskilt av mellomrom og linjeskift  
(ulikt antall på linjene)

Lesing fra fil (beregner sum av alle heltallene):

```
int sum = 0;
while ( leser.hasNextInt() )
    sum += leser.nextInt();
leser.close();
out.println("Sum: " + sum);
```

**OBS!**

hasNextDouble og nextDouble leser desimaltall  
... Utfordring: komma eller punktum i siffersekvensen?

PRG1000B Grunnleggende programmering

Leksjon 6/Kap. 7 – del 1

14

## Strukturerte data

- ulike datatyper på filen

varedata.txt

int	String	double	int
1	Spade	220.50	23
2	Hakke	199.00	17
3	Spett	170.00	58

linje

splitt

1

Spade

220.50

23

int

String

double

int

### «Skilletegnseparert fil» med varedata

- Første linje *kan* ha info om dataene (bare kaste den?)
- Øvrige linjer har skilletegn mellom verdier i ulike datatyper (men, samme oppbygging av hver linje)
- Les en hel linje, splitt den på skilletegnet, og flytt verdiene over i egnede variabler (tabell) → Scanner(linje)

PRG1000B Grunnleggende programmering

Leksjon 6/Kap. 7 – del 1

15

## Strukturerte data - oppsplitting

```

:
File innFil = new File("navn.txt");
Scanner leser = new Scanner(innFil);
while ( leser.hasNextLine() ) {
    String linje = leser.nextLine();
    Scanner linjeleser = new Scanner(linje);
    linjeleser.useDelimiter(";");
    while ( linjeleser.hasNext() ) {
        int heltall = parseInt(linjeleser.next());
        String navn = linjeleser.next();
        double destall = parseDouble(linjeleser.next()); // Kastes bare
        out.println(navn + ", " + heltall);
    }
} // Husk import av Integer og Double
:

```

PRG1000B Grunnleggende programmering

Leksjon 6/Kap. 7 – del 1

16