

Seminar uke 2

IN1010 V23 - Gruppe 9 prosa

srhellst på Discourse
Epost: srhellst@ifi.uio.no



UiO : Universitetet i Oslo

Pensum uke 2

- Samlinger (array, ArrayList, Hashmap)
- Unntakshåndtering
- Manipulere strenger
- Lese fra fil / skrive fra fil
- To-dimensjonale array (ref. oblig 1)
- Datastrukturtegning

Repetisjon: Skrive og kjøre programmer

- All kode ligger i klasser, dvs vi må skrive minst én klasse
- 1 Fil per klasse, klassenavnet nøyaktig likt *Klassenavn.java*
- Må deklarere variabler - variabler kan kun holde denne gitte verdien
- Krøllparantes {} for kodeblokker (klasser, metoder, for-løkker, if-setninger)
- Kjøring av et program starter alltid i en metode som heter **main**
- *Kompilere og kjøre et program:*
 1. *javac Klassenavn.java*
 2. *java Klassenavn*

```
J Program.java > ...
1   public class Program {
2       public static void main(String[] args) {
3           int tall = 10;
4           String tekst = "Hei! Tallet er: ";
5           System.out.println(tekst + tall);
6       }
7   }
```

Repetisjon: Lese bruker input fra terminal

```
J LeseFraTerminal.java > ...
1  import java.util.Scanner;
2
3  public class LeseFraTerminal {
4      public static void main(String[] args) {
5          String nesteOrd, nesteLinje;
6          int nesteInt;
7          boolean harNeste;
8
9          Scanner inputInn = new Scanner(System.in); // Scanner objekt - leser input i terminal
10         System.out.print("Skriv setning: ");
11
12         harNeste = inputInn.hasNext(); // finnes det et ord til i input? - true/false
13         nesteOrd = inputInn.next(); // .next() - neste ord/tegn/tall
14         nesteInt = inputInn.nextInt(); // .nextInt() - neste int
15         nesteLinje = inputInn.nextLine(); // .nextLine() - hele linjen med input / resten av input
16
17         // Eksempel i terminal: "Tall: 8 Resten av linjen"
18         System.out.println(harNeste); // true
19         System.out.println(nesteOrd); // Tall:
20         System.out.println(nesteInt); // 8
21         System.out.println(nesteLinje); // Resten av linjen
22
23         inputInn.close();
24     }
25 }
```

Repetisjon: public/private

- **public**: tilgjengelig overalt
- **private**: tilgjengelig kun innenfor samme klasse

- **class** alene: kode tilgjengelig innenfor samme package/mappe. Kommer oftest til å skrive slik
- **public class**: god praksis å ha **public** foran klassen med **main** metode (men kjører også uten *public* foran når man er mappen)

Repetisjon: Løkker

```
J Lokker.java > ...
1  public class Lokker {
2      public static void main(String[] args) {
3          String[] arrayOrd = {"Hei", "paa", "deg"}; // eller tomt array: new String[3]
4          // For i
5          for (int i = 0; i<arrayOrd.length; i++){
6              System.out.println(i); // 0, 1, 2
7              System.out.println(arrayOrd[i]); // ord fra array
8          }
9          // While
10         int teller = 0;
11         while (teller < arrayOrd.length) { // condition, kjører mens 'true'
12             System.out.println(teller); // 0, 1, 2
13             System.out.println(arrayOrd[teller]); // ord fra array
14             teller++;
15         }
16         // For each
17         for (String ord : arrayOrd){ // definere type for hvert element
18             System.out.println(ord);
19         }
20     }
21 }
```

Samlinger: array - ArrayList - HashMap

array

Fast lengde

```
// --- ARRAY ---  
// uten innhold:  
int[] arrayTall1;  
// [null, null, null]:  
String[] arrayTekst = new String[3];  
// lengde  
int antall = arrayTekst.length; // 3  
// med verdier  
int[] arrayTall2 = {1,2,3};  
// hente element  
int tall = arrayTall2[1]; // 2  
// endre på gitt plass  
arrayTall2[1] = 9; // før 2, nå 9
```

ArrayList (importere)

Kan legge til / fjerne

```
// --- ArrayList ---  
ArrayList<Integer> list = new ArrayList<>();  
list.add(1);  
// hente på index  
int tallArrayList = list.get(0);  
// endre på gitt plass  
list.set(0, 2); // før 1, nå 2  
// fjerne på index  
list.remove(0);  
// lengde  
int str = list.size();
```

HashMap (importere)

Nøkkel (unike) - verdi

```
// --- HashMap ---  
HashMap<String, Integer> map = new HashMap<>();  
// legge til  
map.put("key", 1);  
// hente verdi  
int verdi = map.get("key"); // 1  
// endre på gitt nøkkel  
map.put("key", 2); // før 1, nå 2  
// fjerne element  
map.remove("key");  
// lengde  
int mapStr = map.size();
```

Håndtere unntak - try / catch

- Fanger unntaket der det oppstår
- **try**: kode som man “tester” - kan gi feil
- **catch**: kode som blir utført dersom det er en feil i **try** blokken
- **finally**: kode som blir utført til slutt uavhengig av resultatet i try-catch

```
1 public class TryCatch {
2     public static void main(String[] args) {
3         try {
4             int[] myNumbers = {1, 2, 3};
5             System.out.println(myNumbers[10]);
6         }
7         catch (Exception e) {
8             System.out.println("Something went wrong.");
9         }
10        finally {
11            System.out.println("The 'try catch' is finished.");
12        }
13    }
14 }
```


Håndtere unntak - throws

- **throws**: programmet “kastes” fra metoden og tilbake der metoden kalles
- Bruker med en unntakstype som finnes i Java
- FileNotFoundException, ArrayIndexOutOfBoundsException, ...

```
1 import java.io.File;
2 import java.io.FileNotFoundException;
3 import java.util.Scanner;
4
5 public class LeseFraFil {
6     public static void main(String[] args) throws FileNotFoundException {
7         File fil = new File("minFil.txt"); // hente fil
8         Scanner scanner = new Scanner(fil); // lese filen
9         while (scanner.hasNextLine()){ // kjører så lenge det finnes en li
10             String linje = scanner.nextLine();
11             System.out.println(linje);
12         }
13         scanner.close();
14     }
15 }
```

Manipilere og konverte verdier (strenger)

- Konvertering til/fra String: Integer, Double, Boolean, Char ...

```
minStreng = " 1243 ";  
minStreng = minStreng.trim();  
tall = Integer.parseInt(minStreng);
```

```
int tall = 1234;  
String minStreng = Integer.toString(tall);  
  
double flyttall = 12.34;  
String minStreng2 = Double.toString(flyttall);  
  
System.out.println("minStreng = " + minStreng);  
System.out.println("minStreng2 = " + minStreng2);
```

String nyttige metoder

`s.charAt(pos)` returner karakteren på posisjon `pos` i `s`

`s.equals(s2)` returnerer `true` hvis `s` og `s2` er like tegn for tegn

`s.substring(3, 5)` returnerer kopi av innhold i posisjon 3-4 som ny streng

`split`

`toLowerCase` og `toUpperCase`

`trim`

Lese fra fil

- `public boolean hasNextLine()`
- `public String nextLine()`

```
J LeseFraFil.java > ...
2  import java.io.FileNotFoundException;
3  import java.util.Scanner;
4
5  public class LeseFraFil {
6      public static void main(String[] args) throws FileNotFoundException {
7          File fil = new File("minFil.txt"); // hente fil
8          Scanner scanner = new Scanner(fil); // lese filen
9          while (scanner.hasNextLine()){ // kjører så lenge det finnes en linje til
10             String linje = scanner.nextLine();
11             System.out.println(linje);
12         }
13         scanner.close();
14     }
15 }
```

Skrive til fil

J SkrivetilFil.java > ...

```
1  import java.io.FileNotFoundException;
2  import java.io.PrintWriter;
3
4  public class SkrivetilFil {
5      public static void main(String[] args) throws FileNotFoundException {
6          PrintWriter utFil = new PrintWriter("utfil.txt");
7          utFil.println("Første linje");
8          utFil.println("Legger til en linje");
9          utFil.println("Legger til to linjer\nSiste linje");
10         utFil.close();
11     }
12 }
```

Flerdimensjonal array - struktur Rutenett (o1)

```
public class ToDArray {
    public static void main(String[] args) {
        final int LAND = 7;
        final int MEDALJER = 3;
        int[][] antall = new int[LAND][MEDALJER];

        antall[0][0]++;           // ny gullmedalje for land nr 0

        for (int landNr=0; landNr<LAND; landNr++) {
            for (int medNr=0; medNr<MEDALJER; medNr++) {
                System.out.print(antall[landNr][medNr] + " ");
            }
            System.out.println();
        }
    }
}
```

Datastrukturtegning

- Egentlig ganske enkelt
- Kan gjøres for hånd eller på data
 - draw.io
- Pass på å vise de viktigste tingene
 - Trenger nesten aldri å tegne inn metoder eller andre kodelinjer
- Er du usikker på hvordan noe skal tegnes kan du ta egne valg så lenge du også forklarer dem!



Datastrukturtegning

- Objekter – bokser med avrundede hjørner
- Variabler i rektangler
 - Skriv både variabelnavn og type!
- Piler fra variabler til objekter
- Arrayer tegnes som en rekke bokser
 - Gjerne også med indeksnummer
- Statiske variabler i stiplet objektboks
 - Kalles «Klassedatastruktur <objekttype>»

