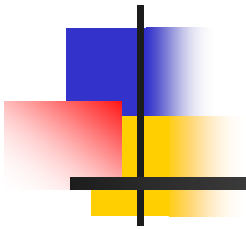
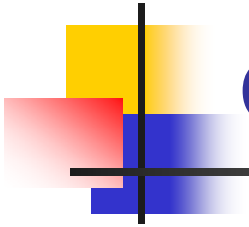


INF1000 - Forelesning 10

- Eksempler på Hashmap
- Oppramstyper
- Innstikksortering
- Javadoc



Oppgave

Anta at du har deklarert en HashMap:

```
HashMap<String,String> cdSamling =
                    new HashMap<String,String>();
```

Du legger inn informasjon om CD'ene dine i HashMapen med platens tittel som nøkkel og artistnavnet som verdi. Eksempel:

```
cdSamling.put("Not going under","Maria
                                  Arredondo");
```

Skriv noen programsetninger som:

- først ber om og leser inn et artistnavn fra tastatur
- deretter går gjennom HashMapen og skriver ut titlene til alle platene du har registrert med denne artisten.

Du kan anta at programsetningene plasseres slik i programmet at de har tilgang til HashMapen **`cdSamling`**.

```
In tast = new In();
System.out.print(”Artistens navn: ”);
String artistnavn = tast.inWord(”\n”);

for( String tittel:cdSamling.keySet() ){
   String navn = cdSamling.get(tittel);
   if (navn.equals(artistnavn))
     System.out.println(tittel);
}
```

Moral når du løper gjennom en HashMap:

- Hvis du bare er ute etter verdien, kan du iterere over values()
- Hvis du er ute etter både nøkkelen og verdien, kan du iterere over keySet() for å finne nøkkelen, og bruke get for å finne verdien.

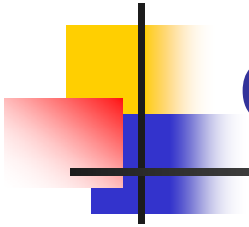
Oppgave

Filen **SpamOrd.txt** inneholder en del *spam-ord* (atskilt av blanke tegn).

Du skal lage et program som leser **SpamOrd.txt** og som deretter leser filen **Epost.txt** og sjekker om denne inneholder noen av spam-ordene. Programmet skriver til slutt ut på skjermen om filen **Epost.txt** inneholdt noen spam-ord eller ikke (du trenger ikke å skrive ut hvilke eller hvor mange spam-ord filen inneholdt).

Du kan anta at **SpamOrd.txt** maksimalt inneholder 200 ord. I denne oppgaven kan du godt legge all programkoden inn i main-metoden.

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

class Spam {
   public static void main (String[] args) {
      HashMap<String,String> h =
            new HashMap<String,String> ();

      In spamfil = new In(”SpamOrd.txt”);
      while (!spamfil.lastItem()) {
         h.put(spamfil.inWord(), null);
      }

      In epostfil = new In(”Epost.txt”);
      boolean funnet = false;
      while (!epostfil.lastItem() && !funnet) {
         if (h.containsKey(epostfil.inWord()) funnet = true;
      }

      if (funnet) System.out.println(”Spam-ord !”);
      else System.out.println(”Ingen spam-ord”);
   }
}
```



Oppgave: Lage et psykolog-program

- Tar inndata – et spørsmål/en setning fra pasienten og svare som en dårlig psykolog
- Ha en fil med enkeltord, og hvis pasienten bruker et av ordene, så svare med tilhørende setning
- Bruke en HashMap for å holde par av ord, setning
- Eks:
  - far, Fortell meg om din far
- Må ha noe å si hvis vi ikke får treff:
  - Så interessant, fortell meg mer om dette.

> hei
Hei, jeg heter Sigmund, hva heter du
> Jeg heter Arne
Så det er altså navnet
> Ja
Fint, skal vi snakke om din barndom?
> Ja og særlig om mor
Fint, skal vi snakke om din barndom?
> Mor sier jeg jo
Lever fortsatt din mor
> nei
Så du mener ikke det
> hun slo meg ofte
Var det ofte at noen slo?
> ganske ofte
Interessant. Fortell mer.
>

Programskisse

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

class Eliza {
  public static void main(String [] args) {

  }
}

class Samtale {
  HashMap hash = new HashMap<String,String>();
  In tast = new In();
  void lesFraFil() {

  }
  void snakk() {

  }
}
```

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

class Eliza {

  public static void main (String [] args) {
     if (args.length !=1) {
       System.out.println(" bruk: >java Eliza <fil-med-ord> ");
     } else {
       Samtale sam = new Samtale();
       sam.lesFraFil(args[0]);
       sam.snakk();
     }
  } // end main
}
```

```
class Samtale {
    HashMap<String,String> hash =
                new HashMap<String,String>();
    In tast = new In();

    void lesFraFil (String filnavn) {
      In fil = new In(filnavn);
        while (!fil.lastItem()) {
           String søkeord = fil.inWord();
           String svar = fil.inLine();
           hash.put(søkeord, svar);
        }
        fil.close();
        System.out.println
           ("Antall ord lest: " + hash.size());
    }
```

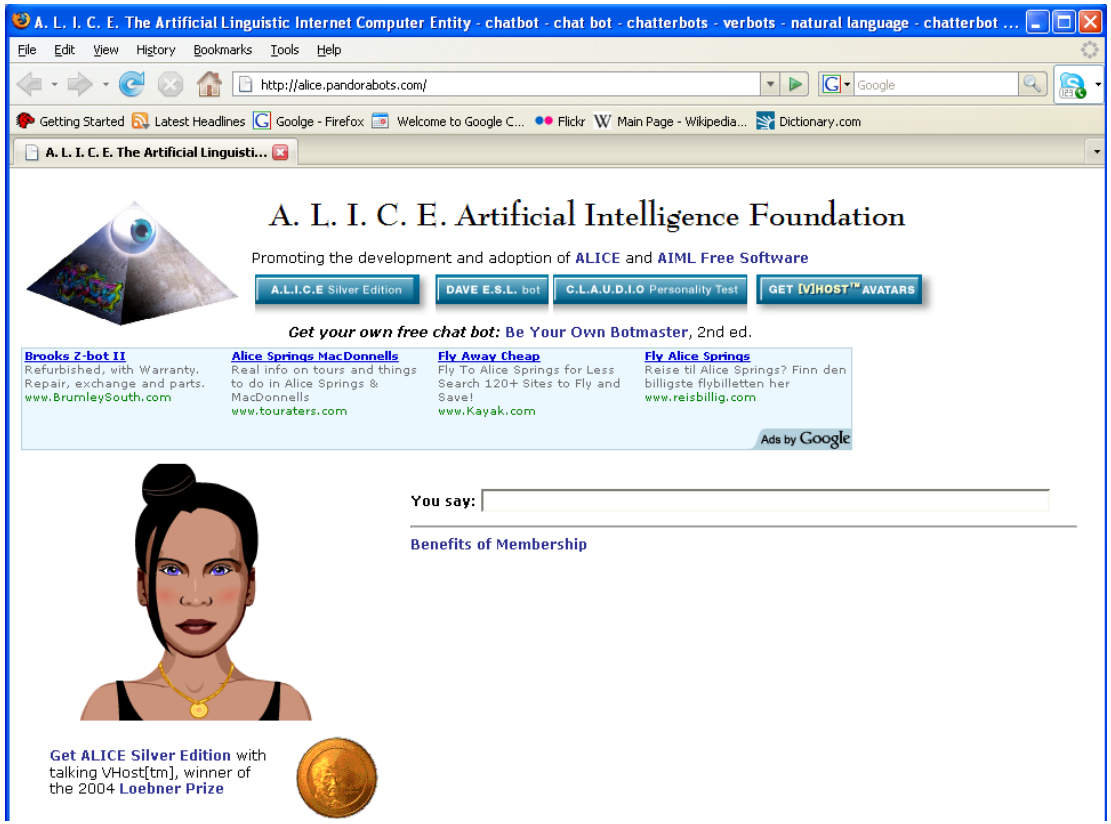
```
void snakk() {
         while (true) {
           System.out.print("> ");
           boolean funnetMatch = false;
           do {
              String ord = tast.inWord().toLowerCase();
              if (hash.containsKey(ord)) {
                  String svar = hash.get(ord);
                  System.out.println(svar);
                  funnetMatch = true;
              }
           } while (tast.hasNextChar() && !funnetMatch);

           if (!funnetMatch) {
              System.out.println("Interessant. Fortell mer.");
           }
           if (tast.hasNextChar()) {
              tast.readLine();  // Tømmer inputbufferet
           }
         }
    } // end snakk
}
```

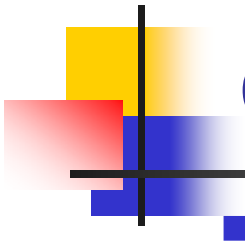
Ordfil.txt

```
far     Fortell meg mer om din far
faren   Hadde du et vanskelig forhold til din far?
slo     Var det ofte at noen slo?
lei     Du sier du er lei deg, hvorfor det
mor     Lever fortsatt din mor
penger  Er du bekymret for om du har nok penger
sint    Hvorfor bruker du 'sint' - er du selv sint
glad    Så bra
ikke    Forklar dette nærmere
vær     Blir du deprimer av dårlig vær
nei     Så du mener ikke det
ja      Fint, skal vi snakke om din barndom?
barn    Har du barn eller barnebarn?
barnebarn Hva heter de
datter  Hvor gammel er hun?
hei     Hei jeg heter Sigmund, hva heter du
jeg     Hvordan føler du deg
heter   Så det er altså navnet
gjenta  Vil du heller snakke om din mor?
```

ALICE: En kunstig intelligens-basert prate-robot

- ALICE = Artificial Linguistic Internet Computer Entity
- http://alice.pandorabots.com





Oppramstyper (`enum`) - motivasjon

- Java-program for å registrere møtedeltakelse
  - Vi trenger f.eks. klassene *Møte* og *Deltaker*
  - En deltaker kan enten
    - *Delta* på møtet
    - *Ikke* delta på møtet
    - *Kanskje* delta på møtet
  - Lagres som *deltakerstatus* for hver *deltaker*
- Hvordan representere i Java?
  - To `boolean`-variable: `delta`, `ikkeDelta`
    - Delta: `delta == true`, `ikkeDelta == false`
    - Ikke delta: `delta == false`, `ikkeDelta == true`
    - Kanskje: `delta == true`, `ikkeDelta == true`
- Tungvint! Bruk heller `enum` i Java...

`enum` – å lage egne oppramstyper

Brukes til å lage 'typer' som har et lite antall verdier

```
enum Status {
        DELTAR,
        DELTAR_IKKE,
        DELTAR_KANSKJE;
}
class EnumEks {
    public static void main(String[] args) {
        Status s = Status.DELTAR;
        System.out.println("Status s er " + s);
        for (Status ss : Status.values()) {
            System.out.println("Status ss er " + ss);
        }
    }
}
```

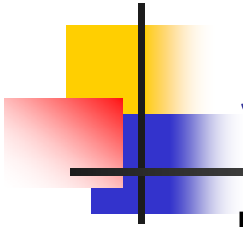
```
Status s er DELTAR
Status ss er DELTAR
Status ss er DELTAR_IKKE
Status ss er DELTAR_KANSKJE
```

‘enum’ kan ha metoder; en enum virker omtrent som en klasse-deklarasjon.

```
public enum Karakter {
    A(6),
    B(5),
    C(4),
    D(3),
    E(2),
    F(0);

    final int verdi;

    boolean erBedre(Karakter k){
        return this.verdi > k.verdi;
    }

    Karakter (int v){
        verdi = v;
    }
  }// end enum Karakter
```

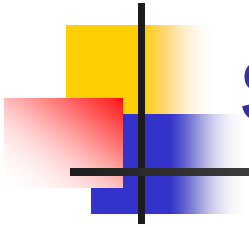
```
class EnumEks2 {
  public static void main(String[] args) {

    Karakter min = Karakter.A, din = Karakter.E;
    System.out.println("Karakteren min  er:" + min);

    if (min.erBedre(din))
      System.out.println("Min karakter:" + min
      + " er bedre enn din " + din);
     }
 } // end class EnumEks2
```

```
Karakteren min  er:A
Min karakter:A er bedre enn din E
```



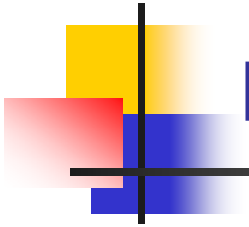
Sortering

- Lære å løse et vanskelig problem
  - Sortering – mange metoder, her *innstikksortering*
  - Sortere hva:
    - Heltall
    - Tekster
- Lære abstraksjon
  - Når vi har løst ett problem, kan lignende problemer løses tilsvarende
- Lære å lage ’proff’ programvare ved å lage en generell klasse (en vektøyboks) for sortering
  - Hvordan deklarere en slik klasse
  - Javadoc – lage dokumentasjon
  - Testing
  - Hvordan utvikle programmet



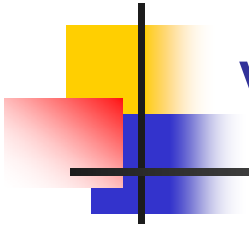
Sortering

- Mange datatyper kan sorteres
  - Tall
  - Tekster (leksikografisk = i samme rekkefølge de ville stått i et leksikon)
  - Tabeller av tekster eller tall
- Vi må ha en algoritme (fremgangsmåte) for sortering
  - Det finns mange metoder for sortering
  - Dere skal lær den som er raskest når vi skal sortere få elementer, si < 50 elementer



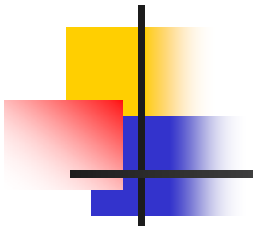
Hvorfor sorterer vi

- For å få noen tall i sortert rekkefølge
  - Eks: lotto-tallene
- Sortere tekster (navnelister)
- Sortere noen opplysninger som hører sammen.
  - Sorterer da på en av opplysningene.
  - Eks. Telefonkatalogen: navn, adresse, telefonnummer sortert på navn



Vi skal først lære å sortere heltall

- Dette skal vi så med minimale endringer bruke til å sortere:
  - String-arrayer (tekster)



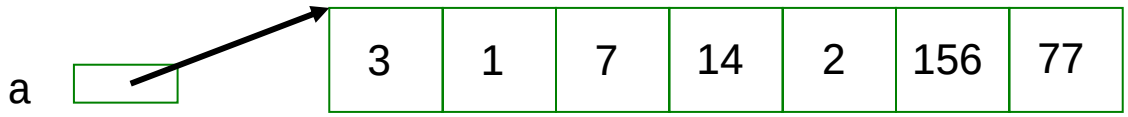
Vi ønsker en klasse med to varianter av sortering: Heltall og tekster

```
public class ISort {

  public static void sorter(int [] a) {

  }
  public static void sorter(String [] a) {

  }
} // end class ISort
```

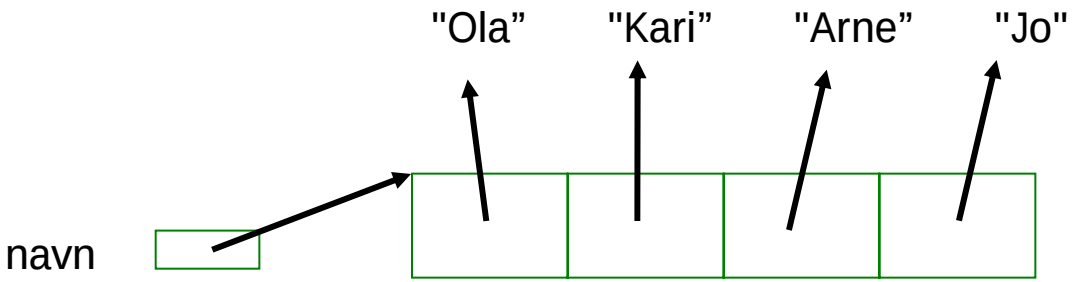
Test-program for sortering

```
class TestInnstikkSortering
{

  public static void  main ( String[] args)  {

     int [] a = {3,1,7,14,2,156,77};
     String [] navn = {"Ola", "Kari", "Arne", "Jo"};

       // sorter heltall - skriv ut
       ISort.sorter(a);
       for (int i = 0; i < a.length; i++)
         System.out.println(”a[" + i +"]= " + a[i]);

         System.out.println("\n Test tekst-sortering:");

       // sorter Stringer - skriv ut
       ISort.sorter(navn);
       for (int i = 0; i < navn.length; i++)
         System.out.println("navn[" + i +"]= " + navn[i]);


  }
}
```

heltalls-array



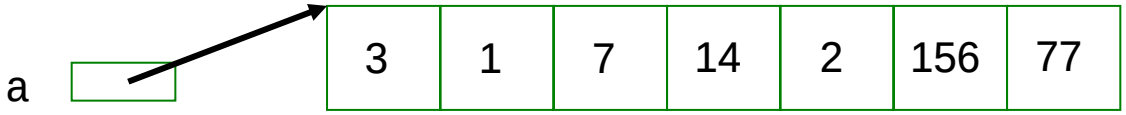
en-dimensjonal
String-array



```
>java InnstikkSortering
a[0]= 3
a[1]= 1
a[2]= 7
a[3]= 14
a[4]= 2
a[5]= 156
a[6]= 77

 Test tekst-sortering:
navn[0]= Ola
navn[1]= Kari
navn[2]= Arne
navn[3]= Jo
```

Test av test-programmet med **tomme** sortering-metoder

En algoritme for å sortere heltall – innstikksmetoden

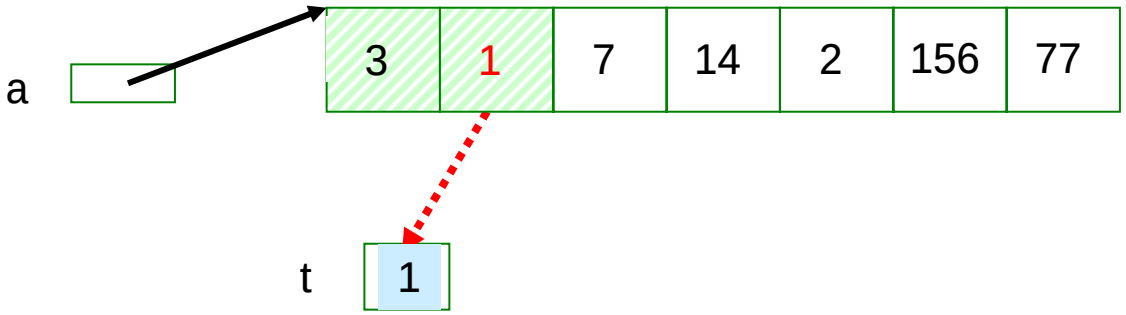


- Se på arrayen *ett for ett* element fra venstre mot høyre
- Sorterer det vi hittil har sett på, ved :
  - Hvis det nye elementet vi ser på **ikke** er sortert i forhold til de vi allerede har sett på:
    - Ta ut dette elementet (gjem verdien i en variabel **`t`**)
    - Skyv på de andre elementene vi her sett på en-etter-en, ett hakk høyreover til elemetet i **`t`** kan settes ned på sortert plass.
    - Da er den delen vi har sortert ett element lenger (fra venstre)
  - Når vi har sett på alle elementene, er hele arrayen sortert
  - Observasjon : Det første elementet er sortert i forhold til seg selv

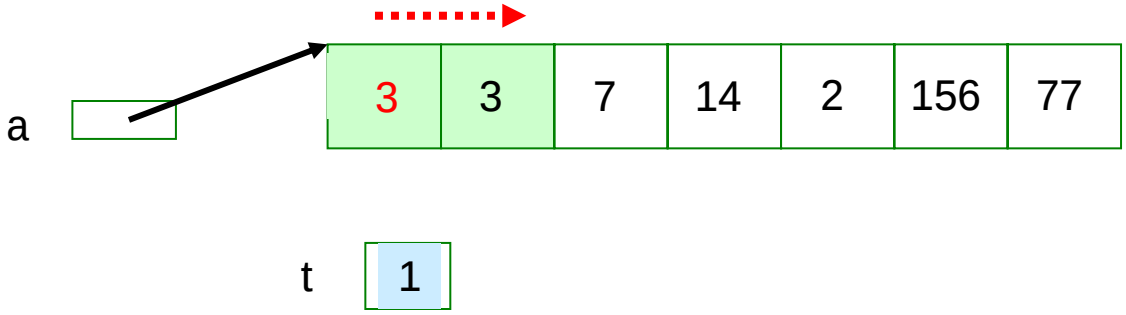
a

Sorter 1 på plass i forhold til 3

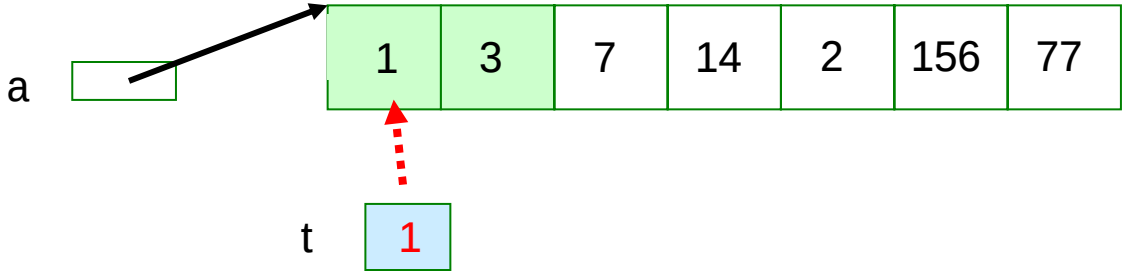
steg 1



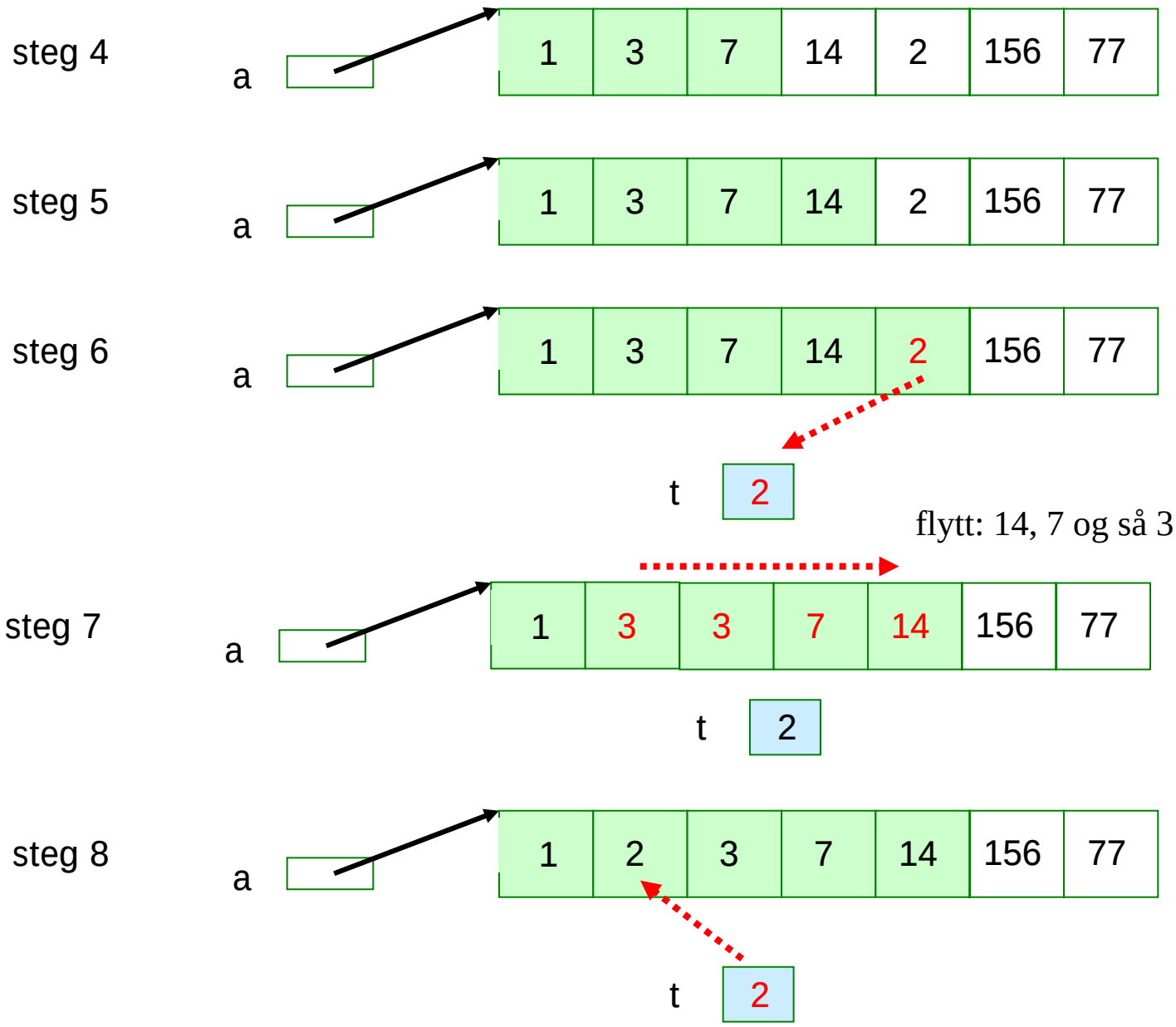
steg 2

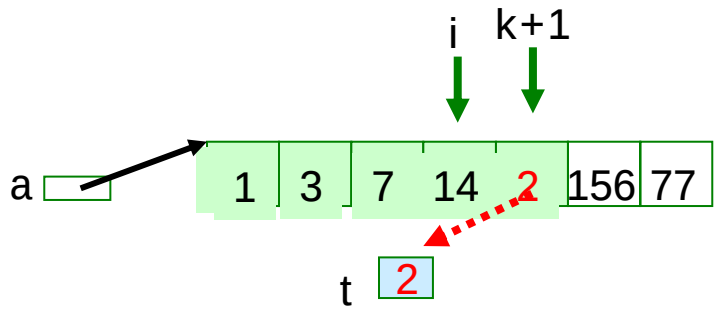


steg 3



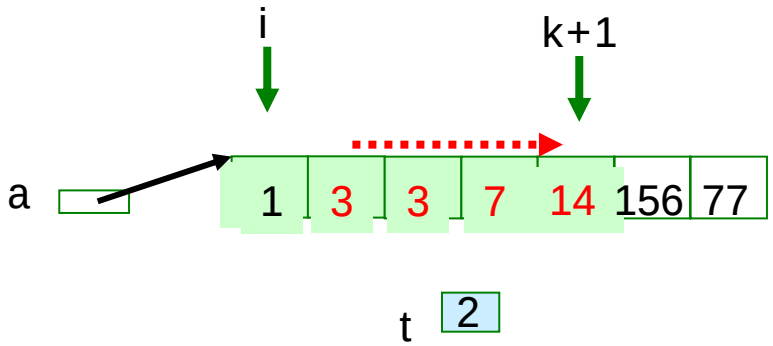
7 og 14 står riktig, Sorter 2 på plass i forhold til : 1,3,7,14



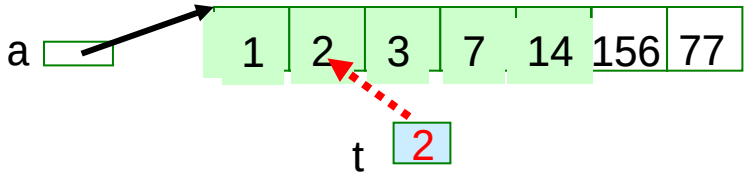


Kode for å flytte **ett** element på plass :

```
// a[k +1] står på
// feil plass, ta den ut
int t = a[k + 1], i = k;
```




```
// skyv a[i] mot høyre ett hakk til
// vi finner riktig plass til t
while (i >= 0 && a[i] > t) {
    a[i + 1] = a[i];
    i--;
}
```



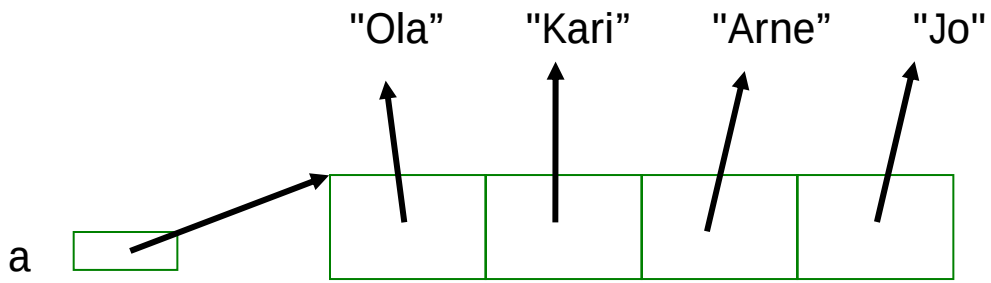

```
// sett t inn på riktig plass
  a[i + 1] = t;
```

```
public class ISort {


  public static void sorter(int [] a) {
       for (int k = 0 ; k < a.length-1; k++) {
         if (a[k] > a[k+1]) {
              // a[k +1 ] står på feil plass, ta den ut
              int t = a[k + 1], i = k;
              // skyv a[i] mot høyre ett hakk til
              // vi finner riktig plass til t
              while (i >= 0 && a[i] > t) {
                     a[i + 1] = a[i];
                     i--;
              }
              // sett t inn på riktig plass
              a[i + 1] = t;
           }
       }
  } // end heltall-sortering
```

```
>java InnstikkSortering
a[0]= 1
a[1]= 2
a[2]= 3
a[3]= 7
a[4]= 14
a[5]= 77
a[6]= 156

 Test tekst-sortering:
navn[0]= Ola
navn[1]= Kari
navn[2]= Arne
navn[3]= Jo
```

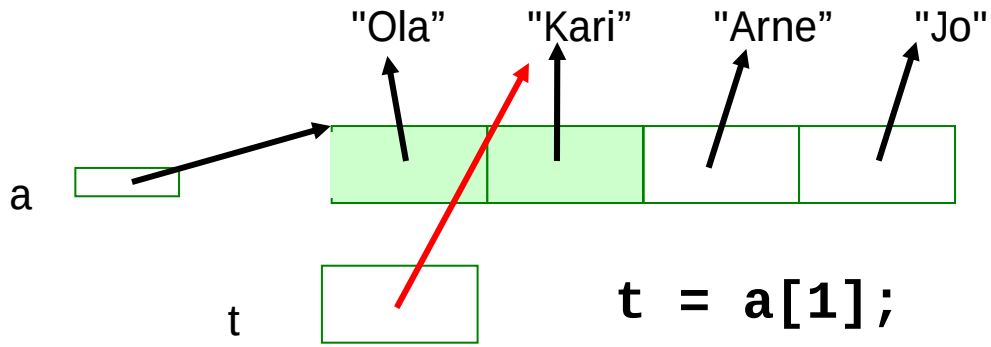
Resultat av sortering med heltalls-metoden kodet, den andre uten kode

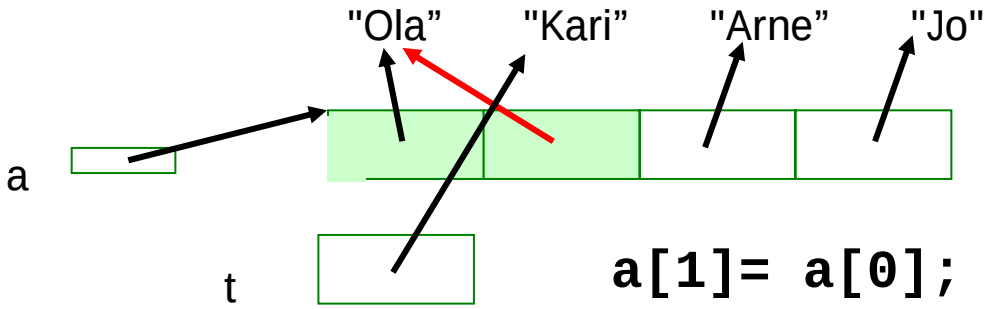
Sortering av tekster (String)

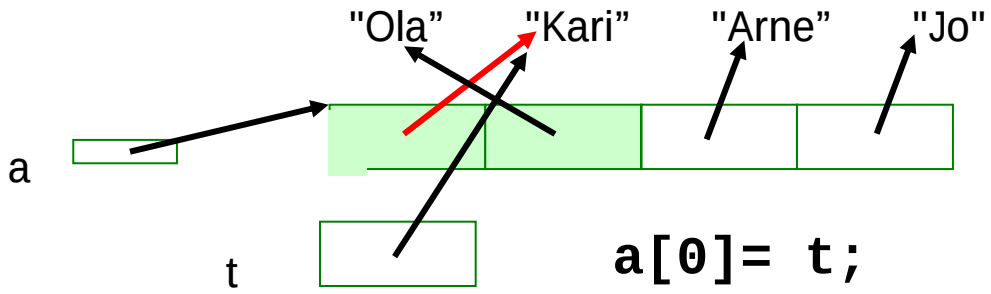


- Vi skal sortere denne ved å bytte om på pekerne (la a[0] peker på ”Arne”,..osv) med innstikkmetoden

Sortere de to første elementene ved å bytte om pekere







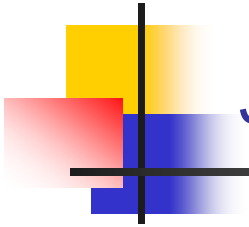
```
public static void sorter(int [] a) {
       // Sorterer heltallsarrayaen 'a'.
       for (int k = 0 ; k < a.length-1; k++) {
          if (a[k] > a[k+1]) {
               int t = a[k + 1];
               int i = k;
               while (i >= 0 && a[i] > t) {
                      a[i + 1] = a[i];
                      i--;
               }
               a[i + 1] = t;
       }  }
  } // end heltall-sortering

  public static void sorter(String [] a) {
       // Sorterer String-arrayen 'a'.
       for (int k = 0 ; k < a.length-1; k++) {
          if( a[k].compareTo(a[k+1]) > 0 ){
               String t = a[k + 1];
               int i = k;
               while (i >= 0 &&  ( a[i].compareTo(t) > 0) ){
                      a[i + 1] = a[i];
                      i--;
               }
               a[i + 1] = t;
       }  }
  } // end String-sortering
```

```
String s = "…";
String t = "…";
s.compareTo(t)
```

- returverdi < 0 hvis s er leksikografisk mindre enn t
- returverdi = 0 hvis s og t er tekstlig like
- returverdi > 0 hvis s er leksikografisk større enn t

```
>java InnstikkSortering
a[0]= 1
a[1]= 2
a[2]= 3
a[3]= 7
a[4]= 14
a[5]= 77
a[6]= 156

 Test tekst-sortering:
navn[0]= Arne
navn[1]= Jo
navn[2]= Kari
navn[3]= Ola
```

Test med heltall og enkel String-sortering kodet



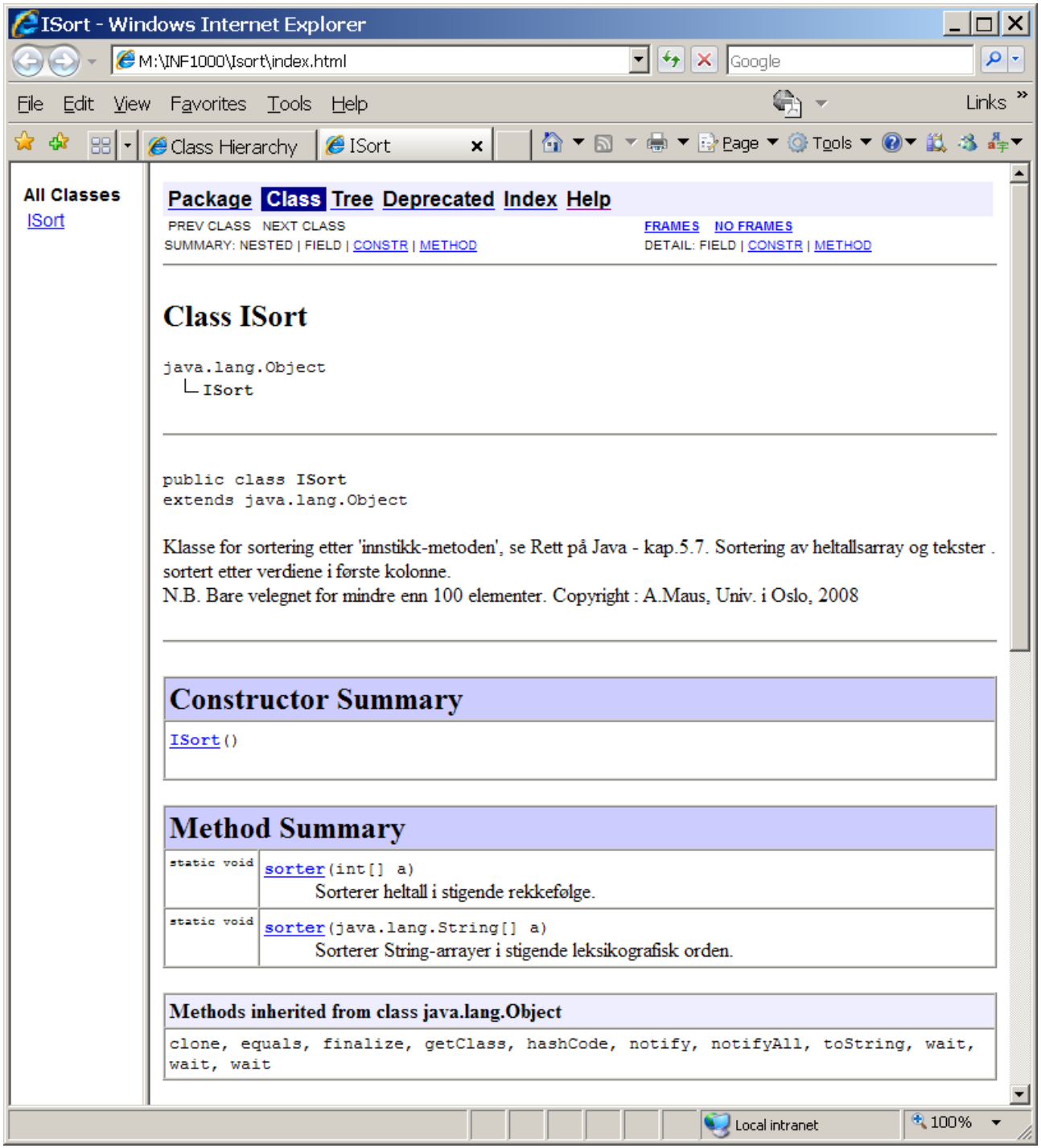
Javadoc – proff dokumentasjon av klassene

- Legg inn spesielle kommentarer i programmet ditt (over hver metode og klasse)
- I disse kommentarene kan man legge HTML-kommandoer (som
 for å få linjeskift)
- Kjør programmet ’javadoc’, og automatisk har du en fin dokumentasjon
- Største fordel: Kode og dokumentasjon vedlikeholdes på samme fil.

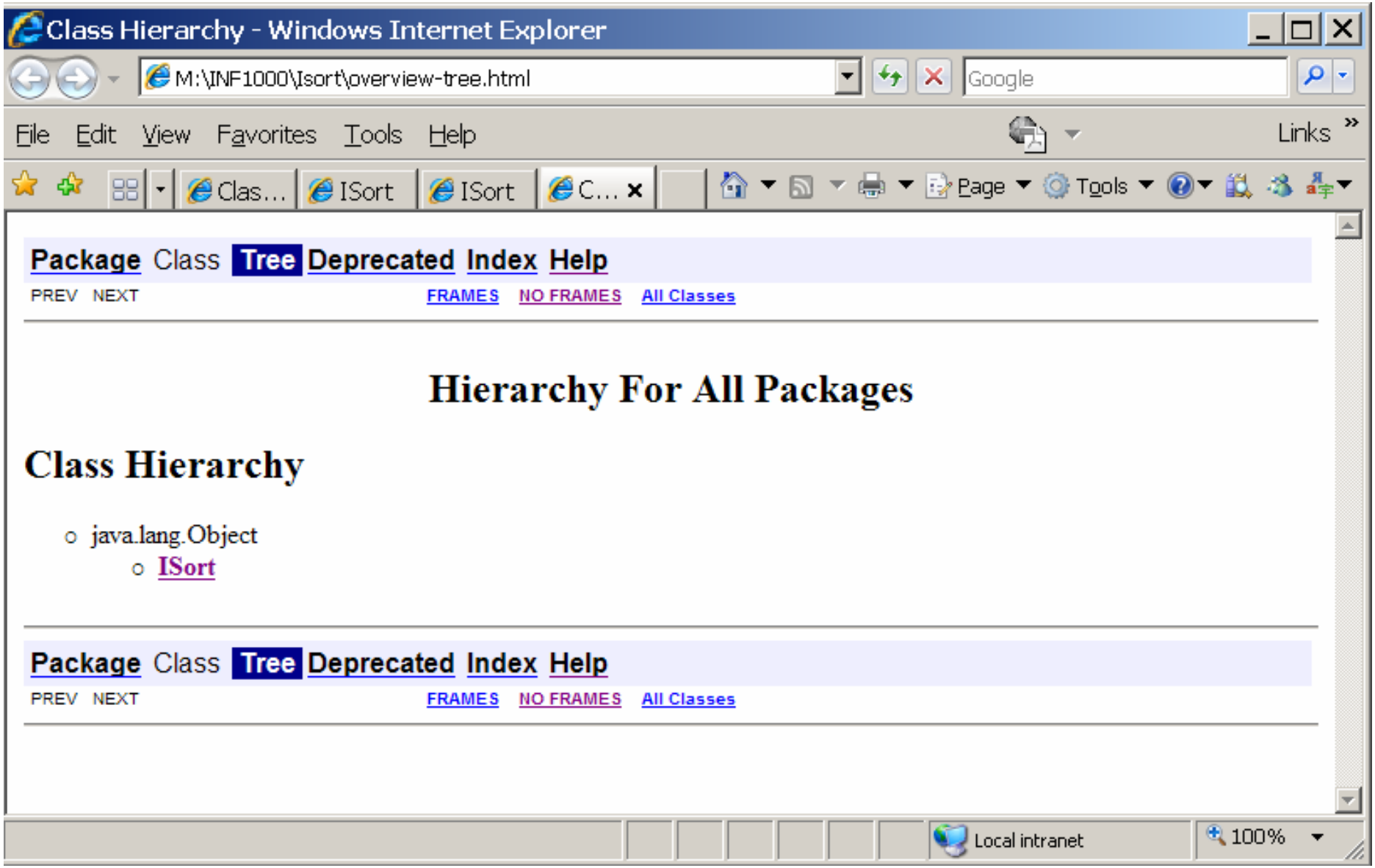
```java
/**
* Klasse for sortering etter 'innstikk-metoden', se
* Rett på Java - kap.5.7.
* Sortering av heltallsarray, tekster og en to-dimensjonal
* tekst-array  sortert etter verdiene i første kolonne.<br>
*
* N.B. Bare velegnet for mindre enn 100 elementer.
*
* Copyright : A.Maus, Univ. i Oslo, 2008
*************************************************************/
public class ISort {

  /**
  * Sorterer heltall i stigende rekkefølge.
  * @param a heltallsarrayen som sorteres. <br>
  * Endrer parameter-arrayen.
  ****************************************/
  public static void sorter(int [] a) {
  }
  /**
  * Sorterer String-arrayer i stigende leksikografisk orden.
  * @param a arrayen som sorteres.<br>
  * Endrer parameter-arrayen
  ****************************************/
  public static void sorter(String [] a) {

  }

} // end class ISort
```

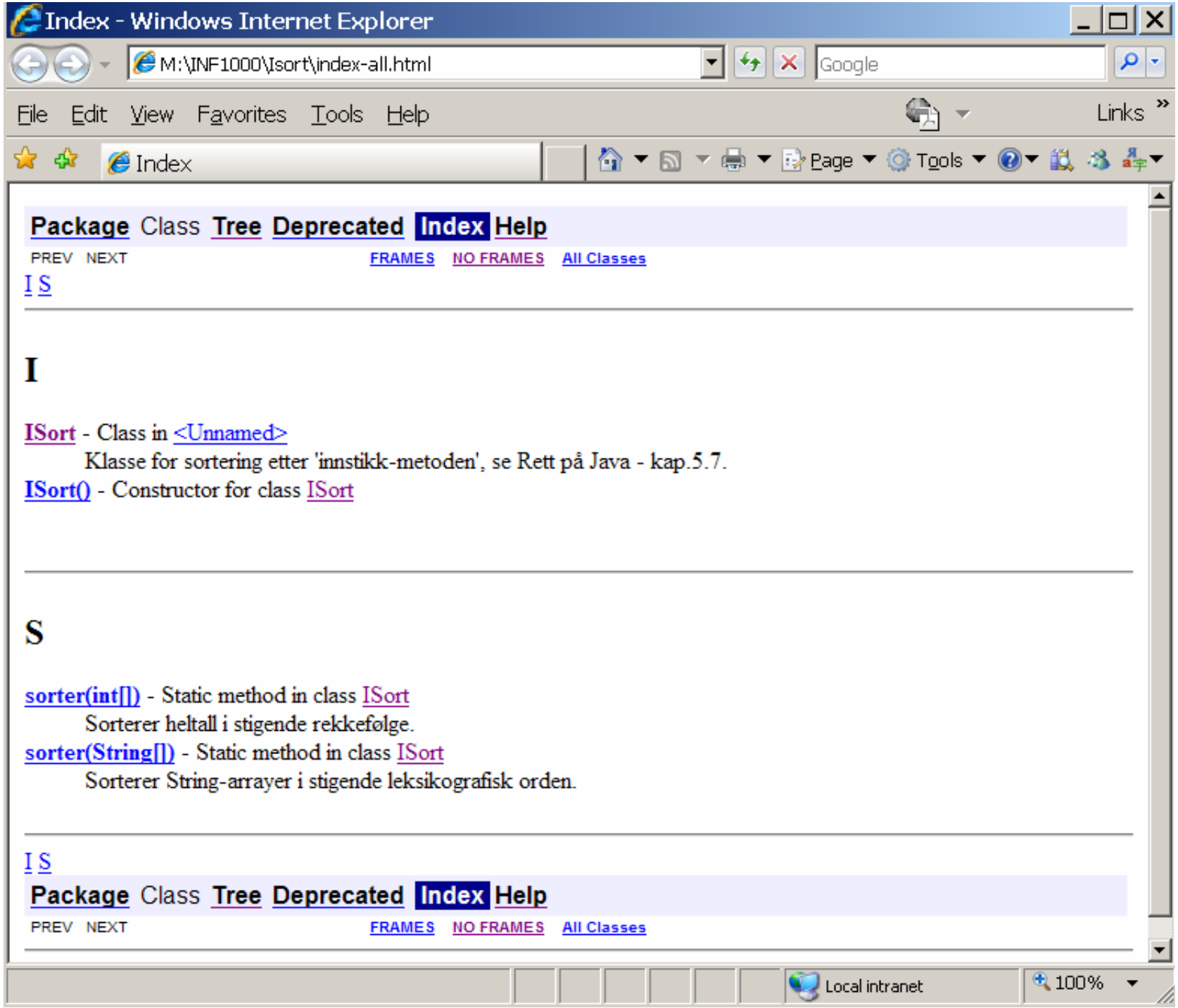
Dokumentasjon av klassen og metodene - javadoc

```
M:\INF1000\Isort>javadoc –package  ISort.java
Loading source file ISort.java...
Constructing Javadoc information...
Standard Doclet version 1.5.0_02
Building tree for all the packages and classes...
Generating ISort.html...
Generating package-frame.html...
Generating package-summary.html...
Generating package-tree.html...
Generating constant-values.html...
Building index for all the packages and classes...
Generating overview-tree.html...
Generating index-all.html...
Generating deprecated-list.html...
Building index for all classes...
Generating allclasses-frame.html...
Generating allclasses-noframe.html...
Generating index.html...
Generating help-doc.html...
Generating stylesheet.css...

M:\INF1000\Isort>
```


ISort - Windows Internet Explorer
M:\INF1000\Isort\index.html
Google
File Edit View Favorites Tools Help
Links
Class Hierarchy
ISort
Page
Tools
All Classes
ISort
Package Class Tree Deprecated Index Help
PREV CLASS NEXT CLASS
FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD
Class ISort
java.lang.Object
└ ISort
public class ISort
extends java.lang.Object
Klasse for sortering etter 'innstikk-metoden', se Rett på Java - kap.5.7. Sortering av heltallsarray og tekster . sortert etter verdiene i første kolonne.
N.B. Bare velegnet for mindre enn 100 elementer. Copyright : A.Maus, Univ. i Oslo, 2008
Constructor Summary
ISort()
Method Summary
static void
sorter(int[] a)
Sorterer heltall i stigende rekkefølge.
static void
sorter(java.lang.String[] a)
Sorterer String-arrayer i stigende leksikografisk orden.
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Local intranet
100%


ISort - Windows Internet Explorer
file:///M:/INF1000/Isort/ISort.html#sorter(int[])
Google
File Edit View Favorites Tools Help
Links
Class Hi...
ISort
ISort
Page
Tools
Method Detail
sorter
public static void sorter(int[] a)
Sorterer heltall i stigende rekkefølge.
Parameters:
a - heltallsarrayen som sorteres Endrer parameter-arrayen.
sorter
public static void sorter(java.lang.String[] a)
Sorterer String-arrayer i stigende leksikografisk orden.
Parameters:
a - arrayen som sorteres Endrer parameter-arrayen
Package Class Tree Deprecated Index Help
PREV CLASS NEXT CLASS
FRAMES NO FRAMES All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD
Local intranet
100%


Class Hierarchy - Windows Internet Explorer
M:\INF1000\Isort\overview-tree.html
Google
File Edit View Favorites Tools Help
Links
Clas... ISort ISort C...
Page Tools
Package Class Tree Deprecated Index Help
PREV NEXT
FRAMES NO FRAMES All Classes
Hierarchy For All Packages
Class Hierarchy
java.lang.Object
ISort
Package Class Tree Deprecated Index Help
PREV NEXT
FRAMES NO FRAMES All Classes
Local intranet
100%


Index - Windows Internet Explorer
M:\INF1000\Isort\index-all.html
Google
File Edit View Favorites Tools Help
Links
Index
Page Tools
Package Class Tree Deprecated Index Help
PREV NEXT
FRAMES NO FRAMES All Classes
I S
I
ISort - Class in <Unnamed>
Klasse for sortering etter 'innstikk-metoden', se Rett på Java - kap.5.7.
ISort() - Constructor for class ISort
S
sorter(int[]) - Static method in class ISort
Sorterer heltall i stigende rekkefølge.
sorter(String[]) - Static method in class ISort
Sorterer String-arrayer i stigende leksikografisk orden.
I S
Package Class Tree Deprecated Index Help
PREV NEXT
FRAMES NO FRAMES All Classes
Local intranet
100%