

Fra Python til Java, del 2

- Hvordan kjøre Java?
 - På Ifis maskiner
 - På egen maskin
- Et eksempel
 - Array-er
 - For-setninger
 - Lesing og skriving
 - Metoder
 - Biblioteket

Å kjøre Java

For å kunne lage og kjøre et Java-program trenger vi:

- et redigeringsprogram for programkode
 - Anbefalt** Atom, Emacs, TextPad og lignende. Bruk gjerne den samme som du brukte til Python i IN1000.
 - Frarådes** Eclipse, NetBeans og andre utviklingsmiljøer.
- Programmet **javac** som kan oversette Java-programmer til class-filer.
- Programmet **java** som kan kjøre class-filer.



Det enkleste kan være å kjøre Java på Ifis maskiner

Kjøre Java på Ifis maskiner

På Ifis maskiner finnes det du trenger.

Hjemmefra

Du kan logge deg inn på en Ifi-maskin hjemmefra:

- Mac: Bruk ssh *navn@login.ifi.uio.no* i et terminalvindu.
- Windows: Bruk Remote Desktop til *windows.ifi.uio.no*.

For å overføre filer anbefales Cyberduck
(<https://cyberduck.io/>).

Java på egen maskin

Det er ganske enkelt å få Java på sin egen maskin.

Versjon

Vi bruker **Java 8** i IN1010.

(Versjon 9 og 10 er også OK, men unngå versjon 11.)

Start med å sjekke maskinen din

Det kan hende du har Java på din maskin allerede. Start et kommandovindu og skriv kommandoene

```
$ java -version  
$ javac -version
```

Hvis begge programmene finnes, og de angir samme versjon av Java, og versjonen er 8, 9 eller 10, er alt i orden.

Hvis bare ett av de to finnes, eller versjonen ikke er 8-10, *må* du fjerne det/dem.

Gamle Java-versjoner

Mange har java (men ikke javac) installert uten å vite om det. Det er fordi mange nettsider krevde Java tidligere:

BankID med Java var ikke helt vellykket:

- Oracle hadde nettopp kjøpt SUN.
- Det var mange småfeil og stadig nye versjoner.
- Nettlesere la inn mange sikkerhetsrestriksjoner.
- Apple sperret Java i Safari og Google i Chrome.

Fra 2014 kunne man bruke BankID uten Java.



Slik installerer du Java på en Mac

Installere Java på en Mac

- 1 Søk på Google etter **download java jdk**. Husk **jdk**!

Java SE Development Kit 8u202

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.86 MB	jdk-8u202-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.75 MB	jdk-8u202-linux-arm64-vfp-hflt.tar.gz
Linux x86	173.08 MB	jdk-8u202-linux-i586.rpm
Linux x86	187.9 MB	jdk-8u202-linux-i586.tar.gz
Linux x64	170.15 MB	jdk-8u202-linux-x64.rpm
Linux x64	185.05 MB	jdk-8u202-linux-x64.tar.gz
Mac OS X x64	249.15 MB	jdk-8u202-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	125.09 MB	jdk-8u202-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	88.1 MB	jdk-8u202-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	124.37 MB	jdk-8u202-solaris-x64.tar.Z
Solaris x64	85.38 MB	jdk-8u202-solaris-x64.tar.gz
Windows x86	201.64 MB	jdk-8u202-windows-i586.exe
Windows x64	211.58 MB	jdk-8u202-windows-x64.exe

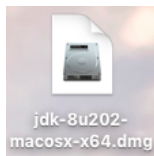
Java SE Development Kit 8u202 Demos and Samples

Last ned versjonen for Mac OS X x64.



Slik installerer du Java på en Mac

3 Dobbelklikk på



og



... og det var det.

Slik installerer du Java på en Mac

Nå kan vi sjekke at alt er i orden:

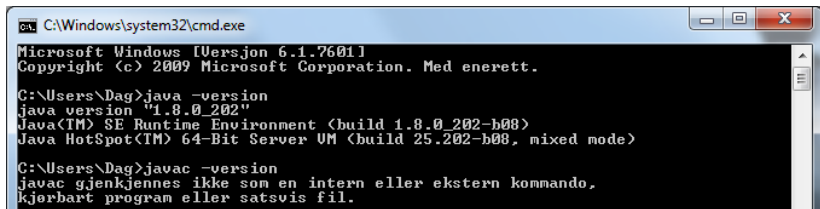
```
Test -- -bash -- 95x19
Dags-iMac:Test dag$ java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
Dags-iMac:Test dag$
Dags-iMac:Test dag$
Dags-iMac:Test dag$ javac -version
javac 1.8.0_202
Dags-iMac:Test dag$
```

og vi kan compilere og kjøre vårt første Java-program:

```
Test -- -bash -- 95x19
Dags-iMac:Test dag$ javac Hallo.java
Dags-iMac:Test dag$ java Hallo
Hallo, alle sammen!
Dags-iMac:Test dag$
```

Og slik installerer du Java på en Windows-maskin

Sjekk først hva som er installert



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versjon 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Med enerett.

C:\Users\Dag>java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)

C:\Users\Dag>javac -version
javac gjenkjennes ikke som en intern eller ekstern kommando,
kjerbart program eller satsvis fil.
```

Her finnes kun **java** men ikke **javac**. Da må **java** avinstalleres.

Og slik installerer du Java på en Windows-maskin

Installere Java på en Windows-maskin

- 1 Søk på Google etter **download java jdk**. Husk **jdk**!

Java SE Development Kit 8u202

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.86 MB	jdk-8u202-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.75 MB	jdk-8u202-linux-arm64-vfp-hflt.tar.gz
Linux x86	173.08 MB	jdk-8u202-linux-i586.rpm
Linux x86	187.9 MB	jdk-8u202-linux-i586.tar.gz
Linux x64	170.15 MB	jdk-8u202-linux-x64.rpm
Linux x64	185.05 MB	jdk-8u202-linux-x64.tar.gz
Mac OS X x64	249.15 MB	jdk-8u202-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	125.09 MB	jdk-8u202-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	88.1 MB	jdk-8u202-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	124.37 MB	jdk-8u202-solaris-x64.tar.Z
Solaris x64	85.38 MB	jdk-8u202-solaris-x64.tar.gz
Windows x86	201.64 MB	jdk-8u202-windows-i586.exe
Windows x64	211.58 MB	jdk-8u202-windows-x64.exe

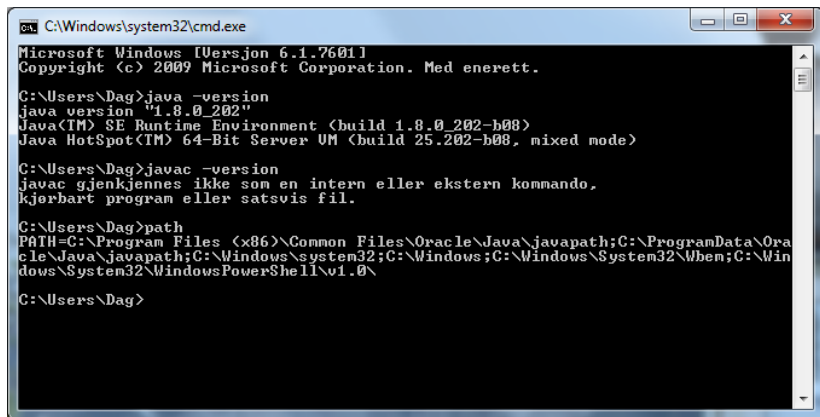
Java SE Development Kit 8u202 Demos and Samples

Last ned versjonen for Windows x64.



Og slik installerer du Java på en Windows-maskin

Men det går ikke alltid glatt!



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versjon 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Med enerett.

C:\Users\Dag>java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)

C:\Users\Dag>javac -version
javac gjenkjennes ikke som en intern eller ekstern kommando,
kjerbart program eller satsvis fil.

C:\Users\Dag>path
PATH=C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\ProgramData\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbin;C:\Windows\System32\WindowsPowerShell\v1.0\

C:\Users\Dag>
```

Kommandovinduet finner stadig bare **java** og ikke **javac**.
Noe må være galt, for eksempel **PATH**.



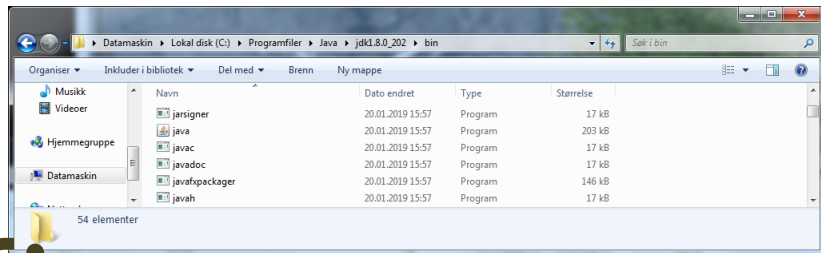
Og slik installerer du Java på en Windows-maskin

Hva er *PATH*?

Alle kommandotolkere har definert en *PATH* som er en liste over hvilke mapper som inneholder nyttige programmer.

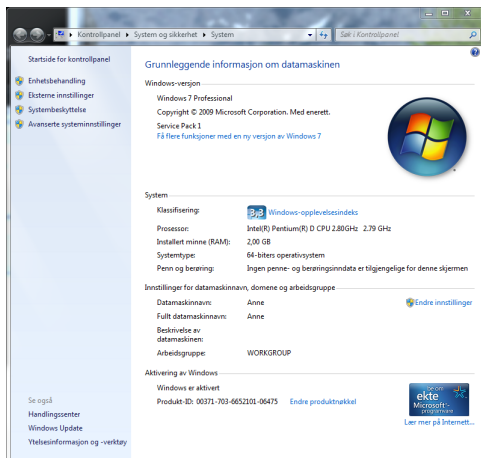
Vår tidligere **java** inkluderte programmet i *PATH*, men avinstalleringen klarte ikke å fjerne den! Da må vi fikse dette selv.

- 1 Finn hvor programmet **javac** befinner seg:



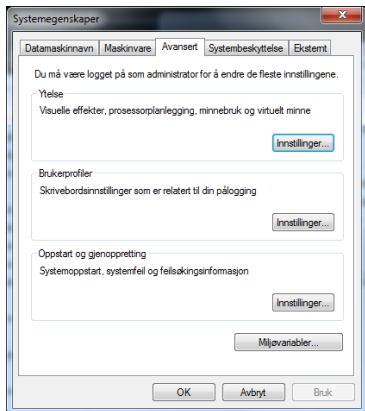
Og slik installerer du Java på en Windows-maskin

Finn kontrollpanelet for systemet:



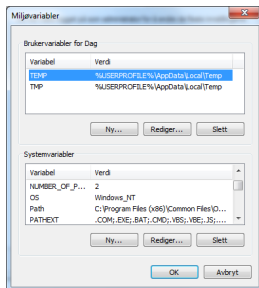
Og slik installerer du Java på en Windows-maskin

Velg «Avanserte systeminnstillinger»:

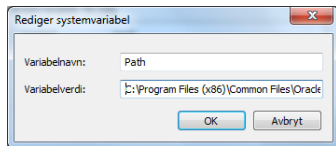


Velg så «Miljøvariabler».

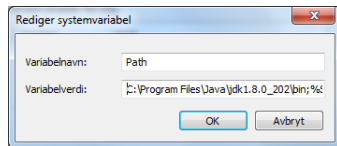
Og slik installerer du Java på en Windows-maskin



Velg «Path» og «Rediger».



til

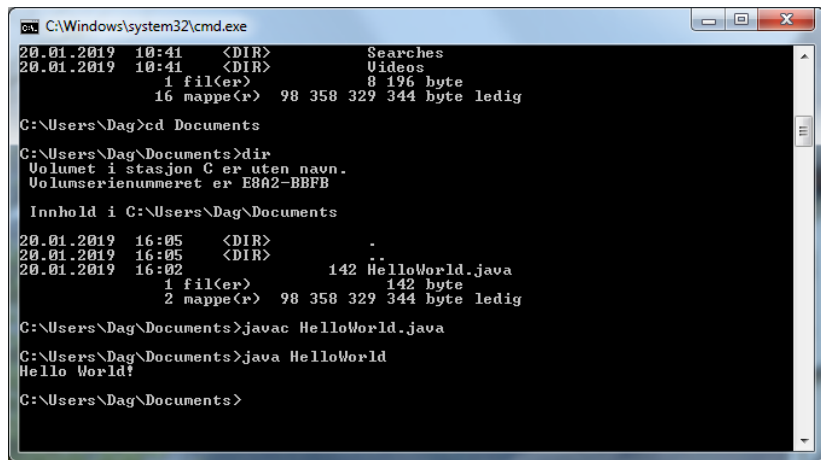


Endre



Og slik installerer du Java på en Windows-maskin

Nå til sist går det bra:



```
C:\Windows\system32\cmd.exe
20.01.2019 10:41 <DIR> Searches
20.01.2019 10:41 <DIR> Videos
                1 fil(er)      8 196 byte
                16 mappe(r)   98 358 329 344 byte ledig

C:\Users\Dag>cd Documents

C:\Users\Dag\Documents>dir
Volumet i stasjon C er uten navn.
Volumserienummeret er E8A2-BBFB

Innhold i C:\Users\Dag\Documents
20.01.2019 16:05 <DIR> .
20.01.2019 16:05 <DIR> ..
20.01.2019 16:02                142 HelloWorld.java
                1 fil(er)      142 byte
                2 mappe(r)   98 358 329 344 byte ledig

C:\Users\Dag\Documents>javac HelloWorld.java

C:\Users\Dag\Documents>java HelloWorld
Hello World!

C:\Users\Dag\Documents>
```

En demonstrasjon

Vi skal programmere Python-programmet (vist forrige tirsdag) på nytt i Java.

```
def finnMedian(a):  
    return sorted(a)[1]  
  
data = [0] * 3  
fil = open("tall.data")  
for i in range(3):  
    data[i] = int(fil.readline())  
print("Medianen er " +  
      str(finnMedian(data)))
```

Hint:

Når vi skal oversette et Python-program til Java, bør vi alltid skrive det på nytt fra bunnen av.

Klasser

I Java må absolutt alt ligge i klasser. Vi må derfor definere en klasse for programmet vårt.

```
class Median {
```

```
def finnMedian(a):  
    return sorted(a)[1]  
  
data = [0] * 3  
fil = open("ta11.data")  
for i in range(3):  
    data[i] = int(fil.readline())  
print("Medianen er " +  
      str(finnMedian(data)))
```

```
}
```



All kjøring starter i main-metoden

For å kunne kjøre programmet trenger vi en **main**-metode:

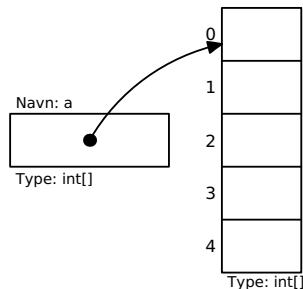
```
def finnMedian(a):  
    return sorted(a)[1]  
  
data = [0] * 3  
fil = open("tall.data")  
for i in range(3):  
    data[i] = int(fil.readline())  
print("Medianen er " +  
      str(finnMedian(data)))  
  
class Median1 {  
  
    public static void main(String[] arg) {  
  
    }  
}
```



Array-er (kontra lister)

I Java har vi **array**-er i stedet for Pythons lister.¹ En array er en datastruktur med mange elementer. Den lagres i etterfølgende celler i minnet.

En array deklarerer med:
`int[] a = new int[5];`



¹I Java-biblioteket finnes ArrayList som ligner ganske mye på Pythons lister; mer om dette senere.

En sammenligning

Pythons lister

- Elementene indekseres fra 0
- + Det sjekkes at indeksen er lovlig
- + Kan endre lengde
- ± Elementene kan være av ulik type
- Mye tregere

Javas array-er

- Elementene indekseres fra 0
- + Det sjekkes at indeksen er lovlig
- Lengden fast når først laget
- ± Alle elementene er av samme type
- + Mye raskere

```
class Median {
```

```
def finnMedian(a):  
    return sorted(a)[1]
```

```
data = [0] * 3  
fil = open("tall.data")  
for i in range(3):  
    data[i] = int(fil.readline())  
print("Medianen er " +  
      str(finnMedian(data)))
```

```
public static void main(String[] arg) {  
    int[] data = new int[3];
```

```
    }  
}
```

Å åpne en fil for lesing

Åpning av fil for lesing

For å kunne lese, benytter vi Scanner-klassen. Ved åpning må vi ta hensyn til hva som skal skje om filen ikke finnes.

```
import java.io.File;
import java.util.Scanner;

class Median {

    def finnMedian(a):
        return sorted(a)[1]

    data = [0] * 3
    fil = open("ta11.data")
    for i in range(3):
        data[i] = int(fil.readline())
    print("Medianen er " +
          str(finnMedian(data)))

    public static void main(String[] arg) {
        int[] data = new int[3];
        Scanner fil = null;
        try {
            fil = new Scanner(new File("ta11.data"));
        } catch (Exception e) {
            System.exit(1);
        }
    }
}
```



Å lese ulike typer verdier

Scanner-objektet inneholder metoder for å lese: `nextInt`, `nextDouble`, `nextLine`, ...

```
import java.io.File;
import java.util.Scanner;
class Median {
    def finnMedian(a):
        return sorted(a)[1]

    data = [0] * 3
    fil = open("tall.data")
    for i in range(3):
        data[i] = int(fil.readline())
    print("Medianen er " +
          str(finnMedian(data)))

    public static void main(String[] arg) {
        int[] data = new int[3];
        Scanner fil = null;
        try {
            fil = new Scanner(new File("tall.data"));
        } catch (Exception e) {
            System.exit(1);
        }
        for (int i = 0; i < 3; i++) { data[i] = fil.nextInt(); }
    }
}
```



For-løkker

For-løkken i Java (og C) er litt uvant:

```
for (int i = 1; i <= 10; i++) {  
    ...  
}
```

med *initiering*, *test* og *inkrementering*.

```
{  
    int i = 1;  
    while (i <= 10) {  
        ...  
        i++;  
    }  
}
```

Dette er bare en alternativ skrivemåte for

Utskrift

Til utskrift bruker vi `println`-metoden. Filen `System.out` er kommandovinduet vi kjører programmet vårt i.

```
import java.io.File;
import java.util.Scanner;
class Median {
def finnMedian(a):
    return sorted(a)[1]
data = [0] * 3
fil = open("tall.data")
for i in range(3):
    data[i] = int(fil.readline())
print("Medianen er " +
      str(finnMedian(data)))
public static void main(String[] arg) {
    int[] data = new int[3];
    Scanner fil = null;
    try {
        fil = new Scanner(new File("tall.data"));
    } catch (Exception e) {
        System.out.println("Finner ikke tall.data");
        System.exit(1);
    }
    for (int i = 0; i < 3; i++) { data[i] = fil.nextInt();
    System.out.println("Medianen er " + finnMedian(data));
    }
}
```



Metoder

Metoder deklarerer som i Python, men vi må angi typen til parametrene og returverdien.

```
def finnMedian(a):  
    return sorted(a)[1]  
  
data = [0] * 3  
fil = open("tall.data")  
for i in range(3):  
    data[i] = int(fil.readline())  
print("Medianen er " +  
      str(finnMedian(data)))  
  
import java.io.File;  
import java.util.Arrays;  
import java.util.Scanner;  
  
class Median {  
    private static int finnMedian(int[] a) {  
        Arrays.sort(a); return a[1];  
    }  
  
    public static void main(String[] arg) {  
        int[] data = new int[3];  
        Scanner fil = null;  
        try {  
            fil = new Scanner(new File("tall.data"));  
        } catch (Exception e) {  
            System.out.println("Finner ikke tall.data");  
            System.exit(1);  
        }  
  
        for (int i = 0; i < 3; i++) { data[i] = fil.nextInt(); }  
        System.out.println("Medianen er " + finnMedian(data));  
    }  
}
```



Biblioteket

Java kommer med et enormt bibliotek; se <https://docs.oracle.com/javase/8/docs/api/index.html?overview-summary.html>.

The screenshot shows the Oracle Java API documentation for the `java.lang.reflect.Array` class. The page is titled "Class Array" and shows the class hierarchy, a method summary table, and the source code for the class.

Method Summary

Modifier and Type	Method and Description
<code>static Object</code>	<code>get(Object array, int index)</code> Returns the value of the indexed component in the specified array object.
<code>static boolean</code>	<code>getBoolean(Object array, int index)</code> Returns the value of the indexed component in the specified array object, as a boolean.
<code>static byte</code>	<code>getByte(Object array, int index)</code> Returns the value of the indexed component in the specified array object, as a byte.
<code>static char</code>	<code>getChar(Object array, int index)</code> Returns the value of the indexed component in the specified array object, as a char.
<code>static double</code>	<code>getDouble(Object array, int index)</code> Returns the value of the indexed component in the specified array object, as a double.
<code>static float</code>	<code>getFloat(Object array, int index)</code> Returns the value of the indexed component in the specified array object, as a float.
<code>static int</code>	<code>getInt(Object array, int index)</code> Returns the value of the indexed component in the specified array object, as an int.
<code>static int</code>	<code>getLength(Object array)</code> Returns the length of the specified array object, as an int.
<code>static long</code>	<code>getLong(Object array, int index)</code> Returns the value of the indexed component in the specified array object, as a long.
<code>static short</code>	<code>getShort(Object array, int index)</code> Returns the value of the indexed component in the specified array object, as a short.
<code>static Object</code>	<code>newInstance(Class<? componentType>, int... dimensions)</code> Creates a new array with the specified component type and dimensions.
<code>static Object</code>	<code>newInstance(Class<? componentType>, int length)</code> Creates a new array with the specified component type and length.
<code>static void</code>	<code>set(Object array, int index, Object value)</code> Sets the value of the indexed component of the specified array object to the specified new value.
<code>static void</code>	<code>setBoolean(Object array, int index, boolean v)</code>

I programmet vårt har vi brukt

java.util.Arrays inneholder diverse nyttige metoder for array-er, blant annet **sort**.

java.io.File benyttes for å lage fil-objekter.

java.util.Scanner gjør det enkelt å lese verdier av ulike typer.

java.lang.System (trenger ikke importeres!) inneholder **exit** avslutter kjøringen (og angir statusverdi: 0=OK) **out** er klassevariabelen som angir hvor vanlig utskrift skal.

Ferdig!

Ferdig!

median.py

```
def finnMedian(a):  
    return sorted(a)[1]  
  
data = [0] * 3  
fil = open("tall.data")  
for i in range(3):  
    data[i] = int(fil.readline())  
print("Medianen er " +  
      str(finnMedian(data)))
```

tall.data

```
0  
-3  
7
```

Kjøring

```
$ javac Median.java  
$ java Median  
Medianen er 0
```

Median.java

```
import java.io.File;  
import java.util.Arrays;  
import java.util.Scanner;  
  
class Median {  
    private static int finnMedian(int[] a) {  
        Arrays.sort(a); return a[1];  
    }  
  
    public static void main(String[] arg) {  
        int[] data = new int[3];  
        Scanner fil = null;  
        try {  
            fil = new Scanner(new File("tall.data"));  
        } catch (Exception e) {  
            System.out.println("Finner ikke tall.data!");  
            System.exit(1);  
        }  
  
        for (int i = 0; i < 3; i++) { data[i] = fil.nextInt(); }  
        System.out.println("Medianen er " + finnMedian(data));  
    }  
}
```

