



DM502

Forelæsning 3

Repetition

- Indlæsning fra tastatur
- Udskrift til skærm
- Beregning af middelværdi
 - Gentagelse med stop-betingelse (`while`)
 - Heltalsdivision
 - Division med nul
- Type-casting (`(double)`)
- Betinget sekvens (`if-else`)
- Test

Repetition

- Datatyper
 - `int` : heltal,
mellem $-2,147,483,648$ og $2,147,483,647$
 - `double` : decimaltal,
mellem $-1.79769 \cdot 10^{308}$ og $1.79769 \cdot 10^{308}$
 - `String` : Tekst

Indhold

- Indlæsning fra en fil
- Fejlhåndtering (exceptions)
- Integer *vs.* int / Double *vs.* double
- Dynamisk datatype: Liste (ArrayList i Java)
- Sortering af lister



Median

- Ligner Middelværdi-programmet meget
- Opgave:
 - Skriv et program der beregner medianen af en liste af heltal
- Specifikation:
 - Inddata: a_1, a_2, \dots, a_N heltal hvor N er ukendt
 - Uddata: Medianen af a_1, a_2, \dots, a_N
- Antagelser:
 - Indlæser tallene fra en fil

Median

- Medianen er “det midterste tal” når tallene skrives op i sorteret rækkefølge (hvis lige antal, så vælg det mindste)
- Antag a_1, a_2, \dots, a_N er sorteret, så er medianen a_i hvor

$$i = \left\lfloor \frac{N + 1}{2} \right\rfloor$$

- Eks: 1, 5, 9, 7, 2 har median 5, da 5 står på index $(5+1)/2 = 3$ i den sorterede liste: 1, 2, 5, 7, 9
- Eks: 7, 1, 3, 5 har median 3, da 3 står på index $(4+1)/2 = 2$ i den sorterede liste 1, 3, 5, 7
- NB: Vi bliver nødt til at huske alle de indlæste tal for at kunne udskrive medianen.

Median

- Algoritme:

Lav en tom liste

Så længe der er tal:

 Læs nyt tal

 Tilføj nyt tal til listen

 Udregn og udskriv medianen



Indlæsning fra fil

- Gøres på samme måde som ved indlæsning fra tastatur
 - Brug `java.util.Scanner`
 - Istedet for `System.in` bruges et objekt af typen `java.io.File`
- Men! Det kan gå galt!
 - Det kan være filen ikke findes
 - Filen kan ikke læses
 - Osv...
- Java kræver at vi forholder os til at det kan gå galt
- Til dette bruges: `try-catch`

try-catch

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

public class OpenFile {
    public static void main( String[] args ) {
        File infile;
        Scanner scan;

        infile = new File( "input.txt" );
        scan = new Scanner( infile );
    }
}
```



try-catch

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

public class OpenFile {
    public static void main( String[] args ) {
        File infile;
        Scanner scan;

        infile = new File( "input.txt" );
        scan = new Scanner( infile );
    }
}
```

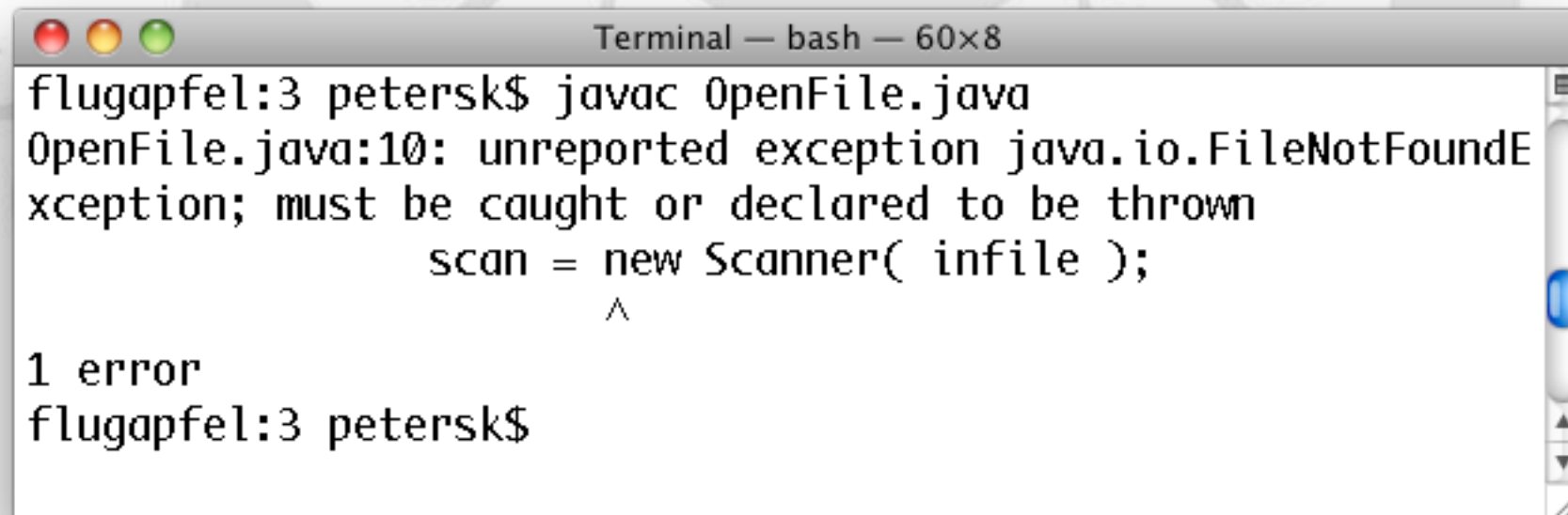


try-catch

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

public class OpenFile {
    public static void main( String[] args ) {
        File infile;
        Scanner scan;

        infile = new File( "input.txt" );
        scan = new Scanner( infile );
    }
}
```



```
Terminal — bash — 60x8
flugapfel:3 petersk$ javac OpenFile.java
OpenFile.java:10: unreported exception java.io.FileNotFoundException;
must be caught or declared to be thrown
        scan = new Scanner( infile );
                        ^
1 error
flugapfel:3 petersk$
```

try-catch

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

public class OpenFile2 {
    public static void main( String[] args ) {
        File infile;
        Scanner scan;

        infile = new File( "input.txt" );
        try {
            scan = new Scanner( infile );
            System.out.println( "Filen blev fundet og åbnet korrekt." );
        } catch( FileNotFoundException fnfe ) {
            System.out.println( "Filen blev ikke fundet!" );
        }
    }
}
```

try-catch

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

public class OpenFile2 {
    public static void main( String[] args ) {
        File infile;
        Scanner scan;

        infile = new File( "input.txt" );
        try {
            scan = new Scanner( infile );
            System.out.println( "Filen blev fundet og åbnet korrekt." );
        } catch( FileNotFoundException fnfe ) {
            System.out.println( "Filen blev ikke fundet!" );
        }
    }
}
```

try-catch

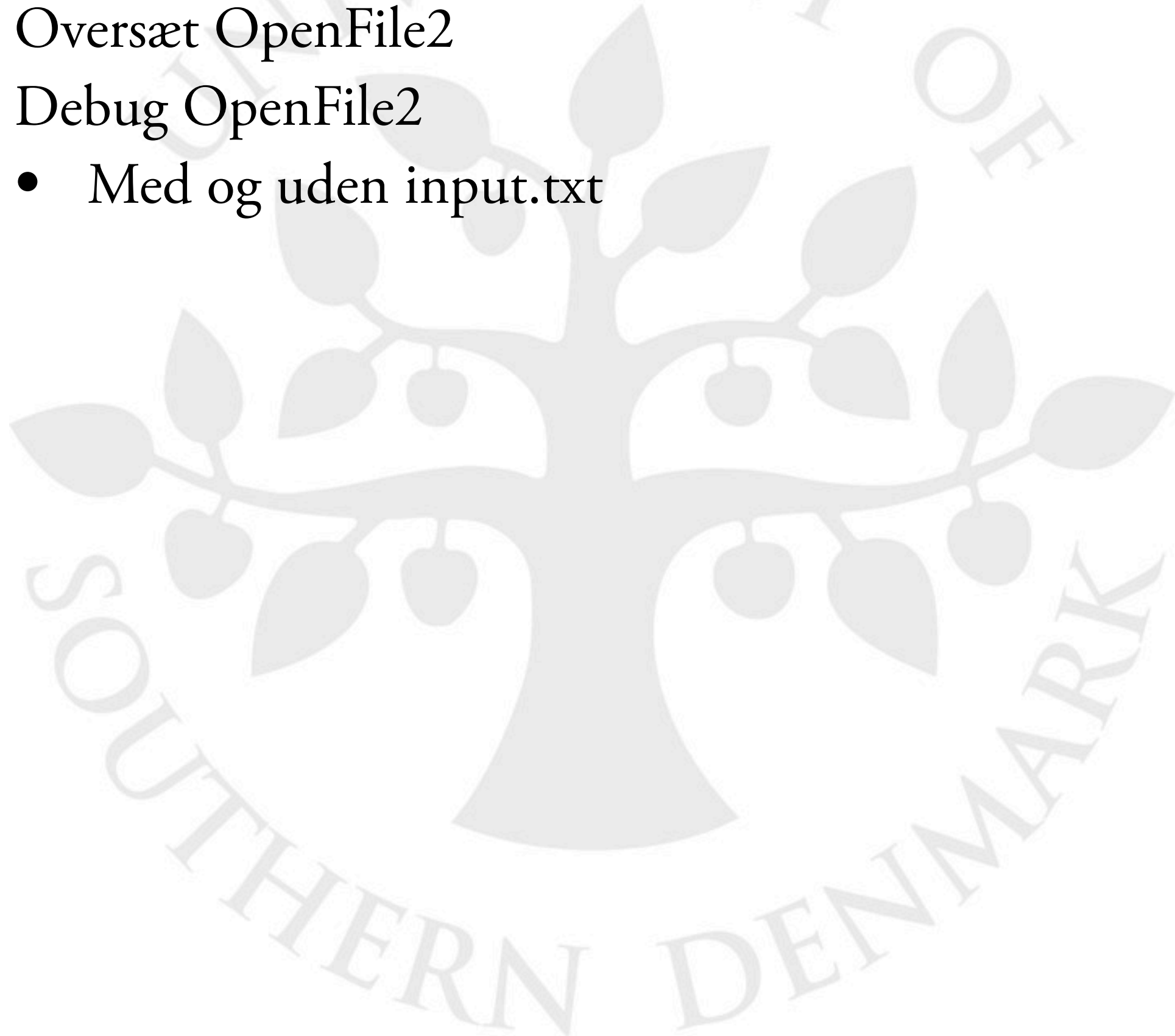
```
import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;

public class OpenFile2 {
    public static void main( String[] args ) {
        File infile;
        Scanner scan;

        infile = new File( "input.txt" );
        try {
            scan = new Scanner( infile );
            System.out.println( "Filen blev fundet og åbnet korrekt." );
        } catch( FileNotFoundException fnfe ) {
            System.out.println( "Filen blev ikke fundet!" );
        }
    }
}
```


try-catch

- Oversæt OpenFile2
- Debug OpenFile2
 - Med og uden input.txt



Indpakning

- I vil ofte komme til at se kode der blander
 - `int` **og** Integer
 - `long` **og** `Long`
 - `float` **of** `Float`
 - `double` **og** Double
 - `char` **og** `Character`
- Forskellen på (f.eks) `int` **og** `Integer` er subtil/spidsfindig/teknisk/forvirrende men ofte af betydning.

Indpakning

- En `int` er en primitiv datatype.
 - Ikke et objekt
 - Har ingen operationer.
 - Er tiltænkt beregninger (der skal gå hurtigt)
 - Fylder lidt i hukommelsen
- En `Integer` er et objekt.
 - Har operationer som `.toString()`
 - Er en pæn indpakning af en `int`

Indpakning

- Hvad er bedst: en `int` eller en `Integer`?
- Hvad er bedst: en `bold` eller en `bold` i en papkasse?
 - Det kommer an på hvad du vil med den!
- Det er heldigvis let at pakke ind og ud
 - ```
int i = ii.intValue();
Integer ii = new Integer(i);
```
- Og ofte vil det kunne gøres automatisk (men det kan nogen gange være farligt)
  - ```
int i = ii;  
Integer ii = i;
```

Lister

- Lister, sekvens af elementer (a_1, a_2, \dots, a_N)
- To typer:
 - Dynamiske: Tillader indsættelser (ArrayList, m.fl.)
 - Statiske: Tillader ikke indsættelser (array)
- Alle lister i Java er 0-indeksret
 - a_0, a_1, \dots, a_{N-1}
 - Det første element står på plads 0
 - Det andet element står på plads 1
 - Osv...
- Vi skal bruge en dynamisk liste, da vi skal indsætte tal løbende.

ArrayList

- ArrayList, en af mange dynamiske lister i Java
- Fordele:
 - Nem at bruge
- Ulemper:
 - Kan ikke indeholde fundamentale typer direkte
 - Med dette gælder for alle “Collections”
 - Vi bliver nødt til at pakke fundamentale typer ind, så de kan lægges i en ArrayList.
- Eksempel:

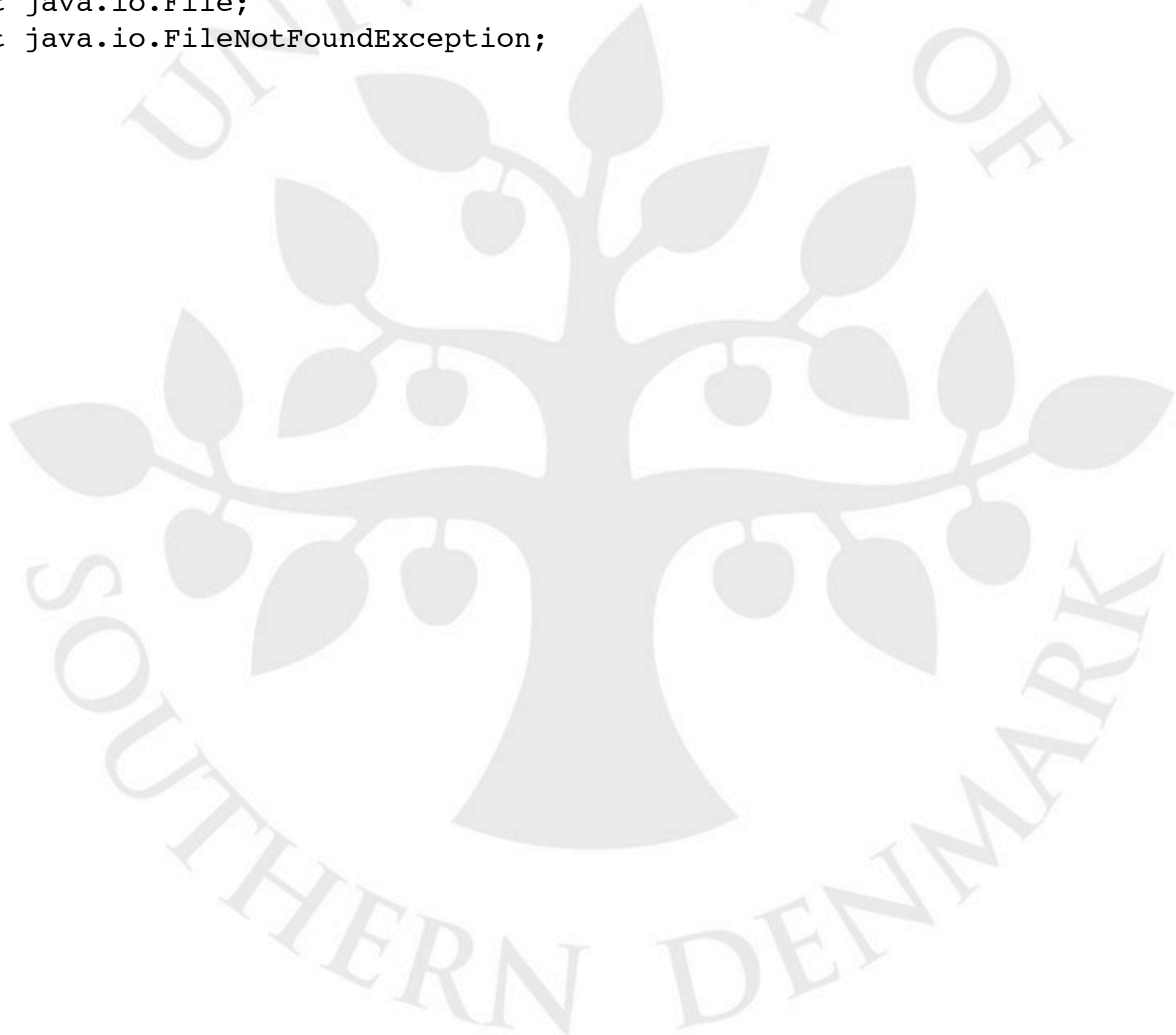
```
ArrayList<Integer> list = new ArrayList<Integer>();  
int i = 42;  
Integer ii = new Integer( i );  
list.add( ii );
```



```
import java.util.Scanner;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.io.File;  
import java.io.FileNotFoundException;
```



```
import java.util.Scanner;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.io.File;  
import java.io.FileNotFoundException;
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;
    }
}
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();
    }
}
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();
    }
}
```




```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }
        }
    }
}
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }
        }
    }
}
```

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }
        }
    }
}
```

Ækvivalent med:

```
Integer i = new Integer(number);
list.add( i );
```



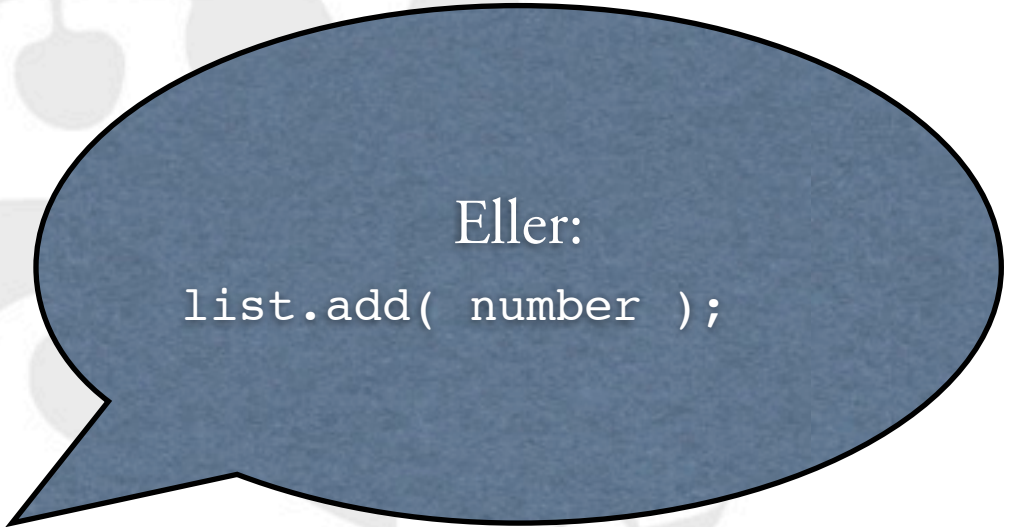
```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }
        }
    }
}
```



Eller:
list.add(number);


```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }
        }
    }
}
```

Eller:

```
list.add( scan.nextInt() );
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }

            Collections.sort( list );
```

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }

            Collections.sort( list );
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }

            Collections.sort( list );

            index = (list.size() + 1) / 2;
            median = list.get( index - 1 );
            System.out.println( "Median: " + median );
        }
    }
}
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }

            Collections.sort( list );

            index = (list.size() + 1) / 2;
            median = list.get( index - 1 );
            System.out.println( "Median: " + median );
        }
    }
}
```

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }

            Collections.sort( list );

            index = (list.size() + 1) / 2;
            median = list.get( index - 1 );
            System.out.println( "Median: " + median );
        } catch( FileNotFoundException fnfe ) {
            System.out.println( "Fejl: Filen blev ikke fundet." );
        }
    }
}
```



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.io.FileNotFoundException;

public class Median {
    public static void main( String[] args ) {
        File infile;
        int number, index;
        Integer median;
        Scanner scan;
        ArrayList<Integer> list;

        infile = new File( "input.txt" );
        list = new ArrayList<Integer>();

        try {
            scan = new Scanner( infile );

            while( scan.hasNextInt() ) {
                number = scan.nextInt();
                list.add( new Integer(number) );
            }

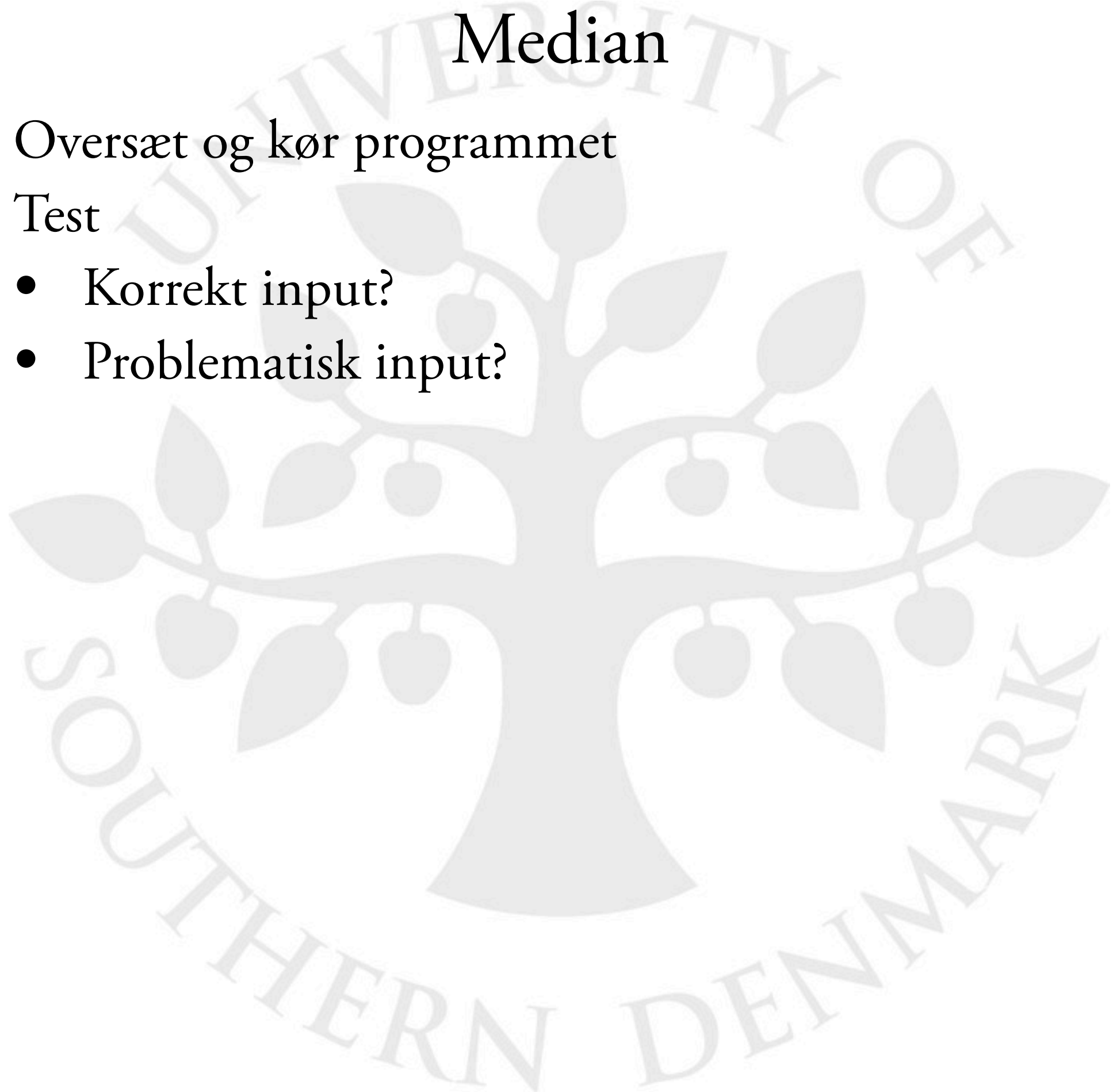
            Collections.sort( list );

            index = (list.size() + 1) / 2;
            median = list.get( index - 1 );
            System.out.println( "Median: " + median );
        } catch( FileNotFoundException fnfe ) {
            System.out.println( "Fejl: Filen blev ikke fundet." );
        }
    }
}
```



Median

- Oversæt og kød programmet
- Test
 - Korrekt input?
 - Problematisk input?



Test med korrekt input

- Eksempel 1:
 - input.txt: 7, 1, 9, 5, 2 (et tal på hver linje)
 - Forventet output: 5
 - Faktisk output: 5
 - Konklusion: Testen forløb som forventet



Test med korrekt input

- Eksempel 2:
 - input.txt: 7, 1, 3, 5 (et tal på hver linje)
 - Forventet output: 3
 - Faktisk output: 3
 - Konklusion: Testen forløb som forventet



Test med problematisk input

- Eksempel 1:
 - Input.txt: (tom fil)
 - Forventet output: ? (medianen ikke defineret)
 - Faktisk output:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1
at java.util.ArrayList.get(ArrayList.java:324)
at Median.main(Median.java:29)
 - Konklusion: Programmet går ned med en fejl. Burde have givet en fornuftig fejlmeddelse i stedet (“Medianen er ikke defineret”)

Median

- Fejlretning: To muligheder
 - Check om der er tal i listen
 - Fang (catch) den exception der opstår

```
median = list.get( index - 1 );  
System.out.println( "Median: " + median );
```



```
try {  
    median = list.get( index - 1 );  
    System.out.println( "Median: " + median );  
} catch( ArrayIndexOutOfBoundsException aioobe ) {  
    System.out.println( "Medianen er udefineret." );  
}
```