

Generiske typer i Java

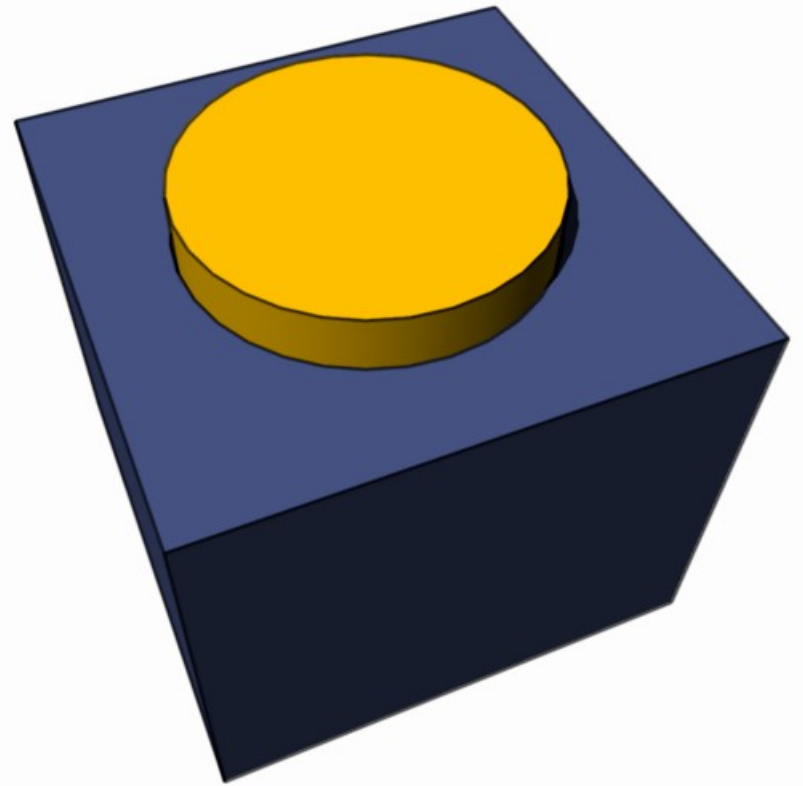
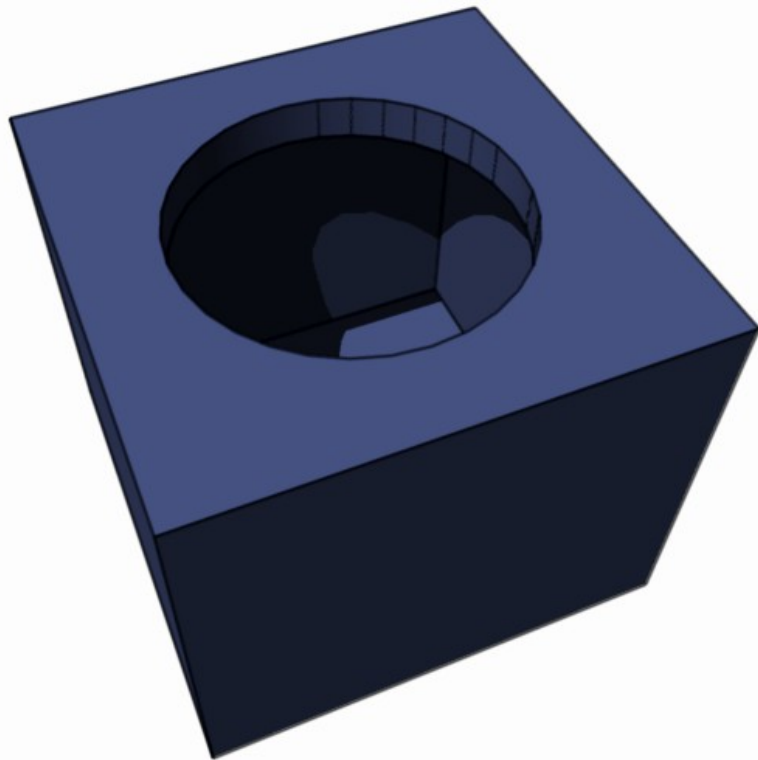
En kort forklaring av Javas «Generics»:

Sigmund Hansen <sigmunha@student.uio.no>

Ikke-generiske klasser:

```
class CylinderBox {  
    Cylinder cyl;  
  
    public boolean insert(Cylinder cyl) {  
        this.cyl = cyl;  
    }  
  
    public Cylinder takeOut() {  
        return cyl;  
    }  
}
```

CylinderBox



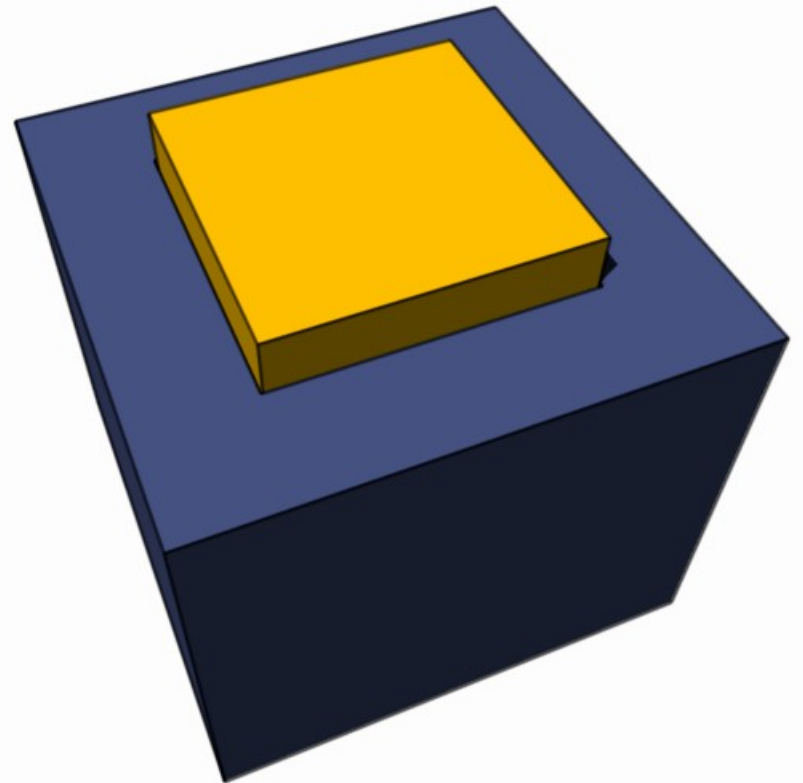
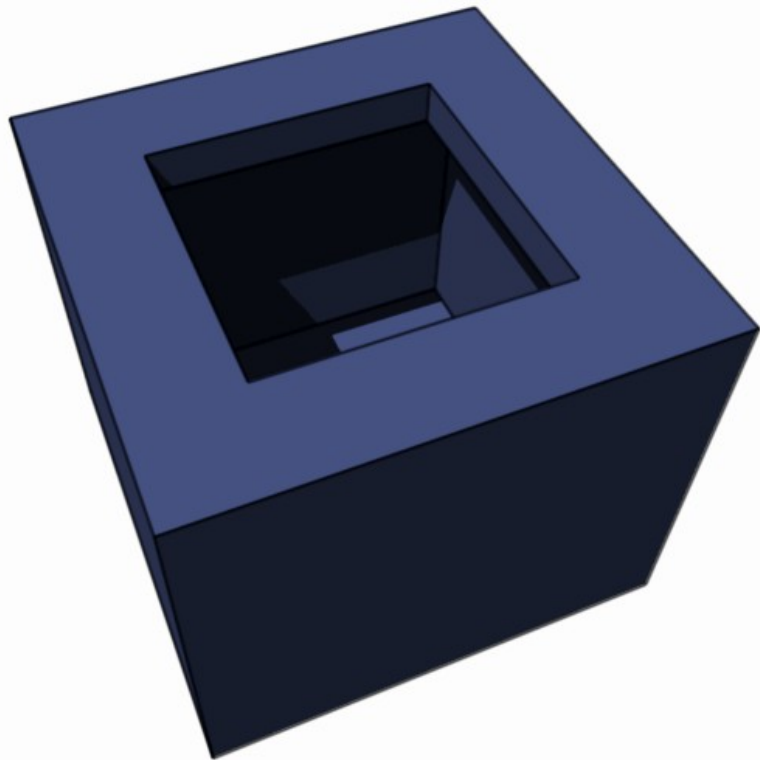
Flere ikke-generiske klasser:

```
class CubeBox {  
    Cube cube;  
  
    public boolean insert(Cube cube) {  
        this.cube = cube;  
    }  
  
    public Cube takeOut() {  
        return cube;  
    }  
}
```

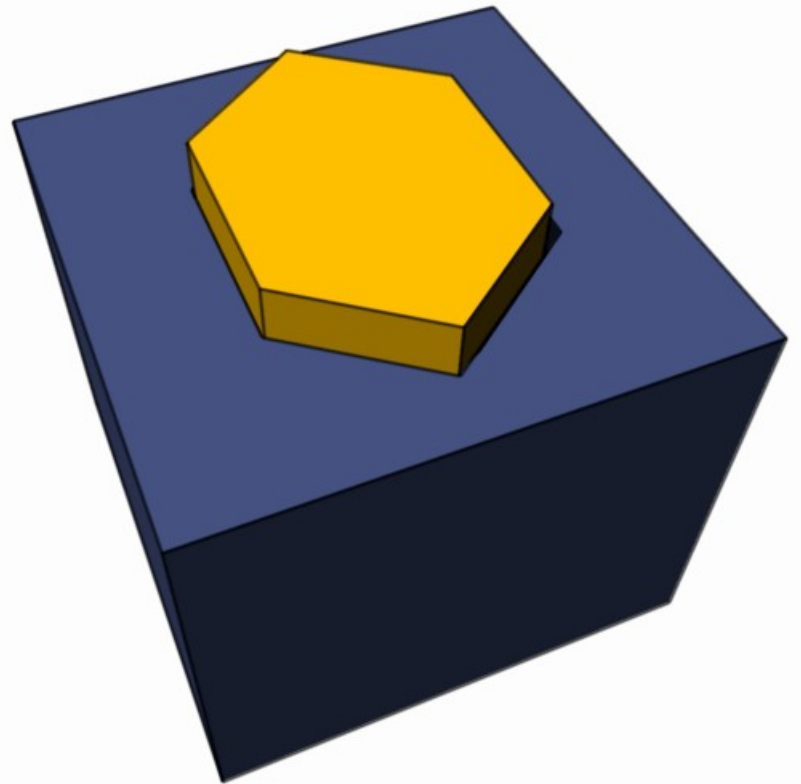
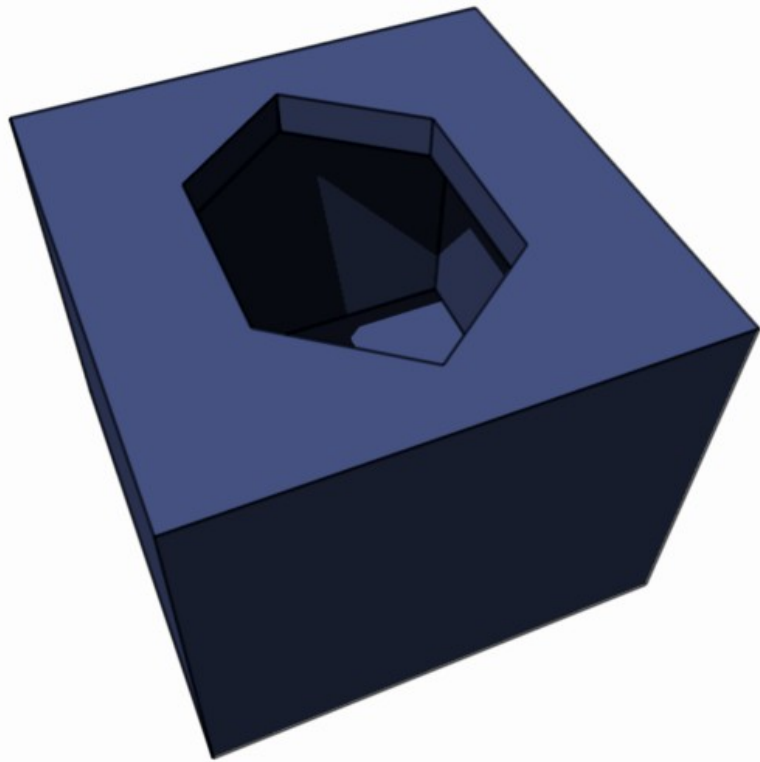
Flere ikke-generiske klasser:

```
class HexBox {  
    Hex hex;  
  
    public boolean insert(Hex hex) {  
        this.hex = hex;  
    }  
  
    public Hex takeOut() {  
        return hex;  
    }  
}
```

CubeBox



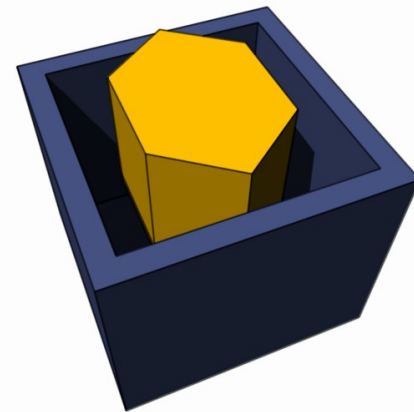
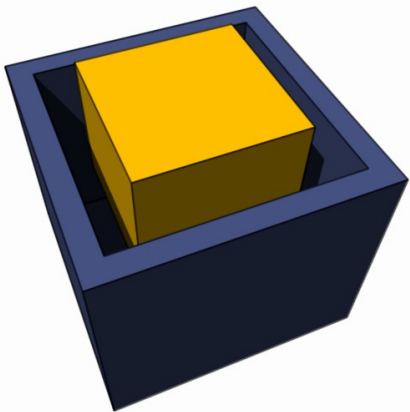
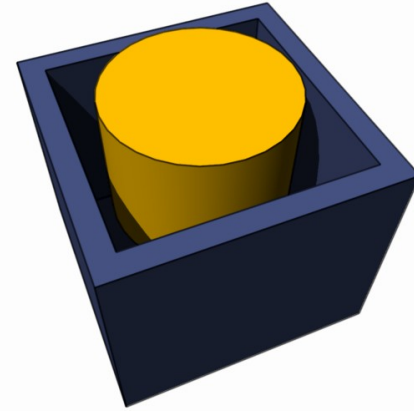
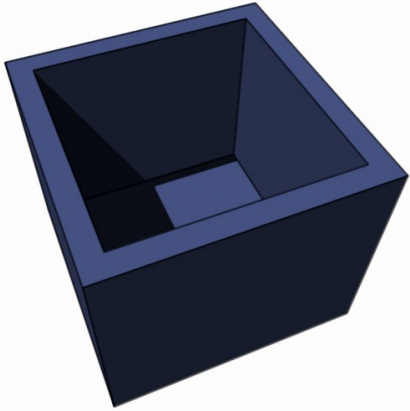
HexBox



En ikke-generisk klasse som tar alt:

```
class AnythingBox {  
    Object item;  
  
    public boolean insert(Object item) {  
        this.item = item;  
    }  
  
    public Object takeOut() {  
        return item;  
    }  
}
```


AnythingBox



AnythingBox i bruk:

```
AnythingBox box = new AnythingBox();  
box.insert(new Cylinder());  
Object item = box.takeOut();
```

```
box.insert(new Cube());  
Object item2 = box.takeOut();
```

```
box.insert(new Hex());  
Object item3 = box.takeOut();
```

```
System.out.printf("%s - %s - %s%n",  
                  item.getClass().getName(),  
                  item2.getClass().getName(),  
                  item3.getClass().getName());
```

En generisk klasse:

```
class Box<T> {  
    T item;  
  
    public boolean insert(T item) {  
        this.item = item;  
    }  
  
    public T takeOut() {  
        return item;  
    }  
}
```

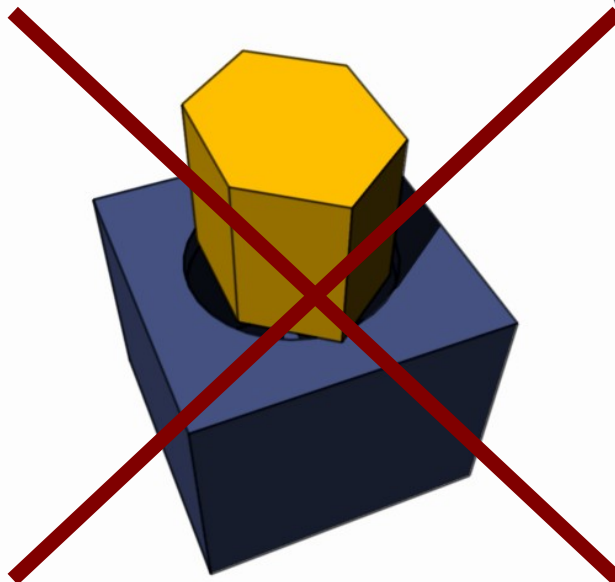
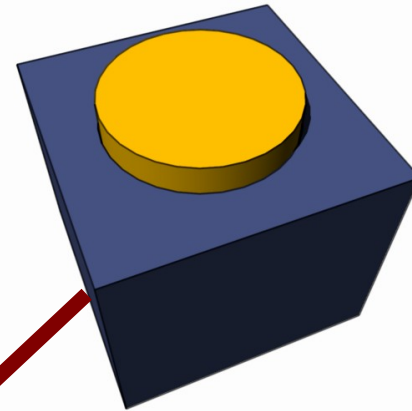
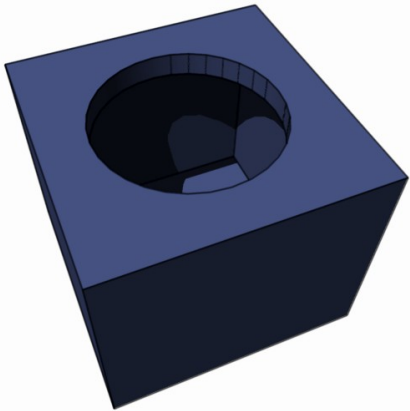
Generisk Box i bruk:

```
Box<Cylinder> box = new Box<Cylinder>();  
box.insert(new Cylinder());  
Cylinder cyl = box.takeOut();
```

```
box.insert(new Hex()); ← Gir kompilatorfeil:
```

insert(Cylinder) in Box<Cylinder> cannot be applied to (Hex)

Box<Cylinder>



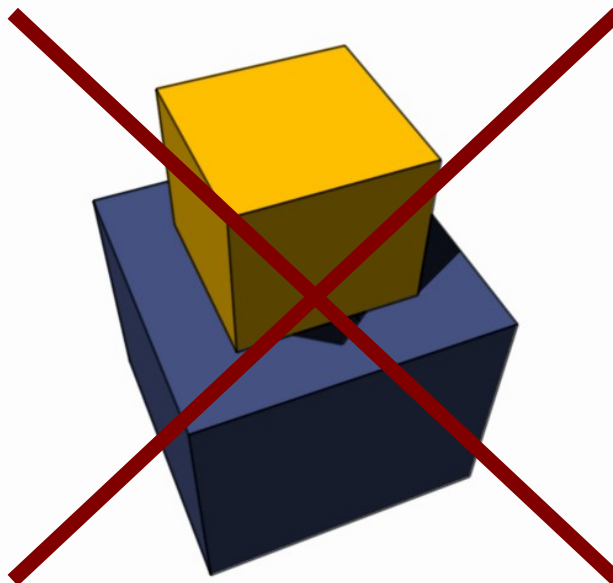
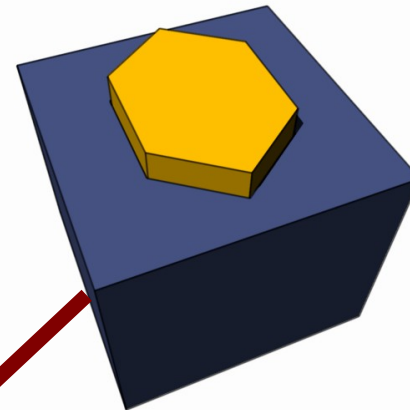
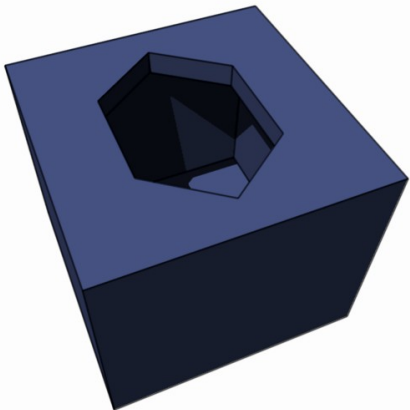
Generisk Box i bruk:

```
Box<Hex> box = new Box<Hex>();  
box.insert(new Hex());  
Hex hex = box.takeOut();
```

```
box.insert(new Cube()); ← Gir kompilatorfeil:
```

insert(Hex) in Box<Hex> cannot be applied to (Hex)

Box<Hex>



Generisk Box i bruk:

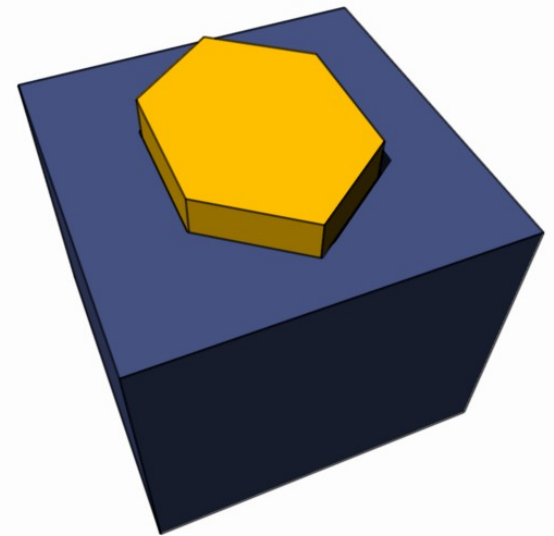
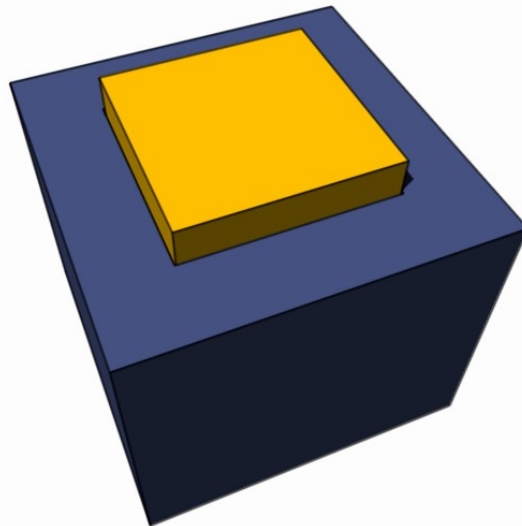
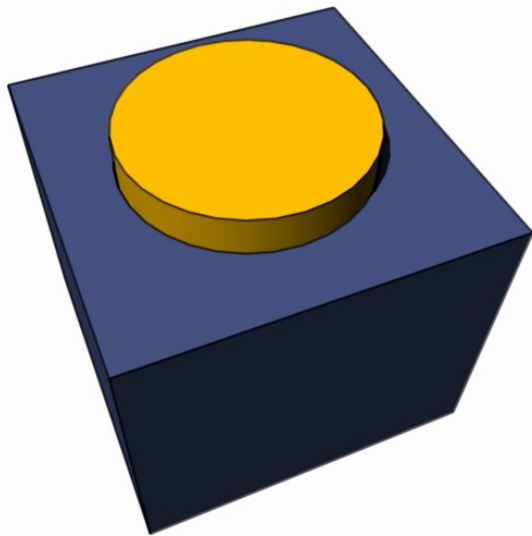
```
Box<Cylinder> cylBox = new Box<Cylinder>();  
Box<Cube> cubeBox = new Box<Cube>();  
Box<Hex> hexBox = new Box<Hex>();
```

```
cylBox.insert(new Cylinder());  
Cylinder cyl = cylBox.takeOut();
```

```
cubeBox.insert(new Cube());  
Cube cube = cubeBox.takeOut();
```

```
hexBox.insert(new Hex());  
Hex hex = hexBox.takeOut();
```


Forskjellige Box-objekter



Generisk Box som tar alt:

```
Box anythingBox = new Box();  
box.insert(new Cylinder());  
Object item = box.takeOut();
```

```
box.insert(new Cube());  
Object item2 = box.takeOut();
```

```
box.insert(new Hex());  
Object item3 = box.takeOut();
```

```
System.out.printf("%s - %s - %s%n",  
                  item.getClass().getName(),  
                  item2.getClass().getName(),  
                  item3.getClass().getName());
```

Advarsel:

Man bør alltid definere en type!!!

Ellers ender man opp med å ikke vite hva som finnes i objektene man lager av den generiske klassen.

Mer info om generiske typer:

Hvis du vil vite mer om generiske typer i Java, les disse:

- <http://download.oracle.com/javase/tutorial/java/generics/index.html>
 - Enkel innføring i generiske typer
- <http://java.sun.com/j2se/1.5/pdf/generics-tutorial.pdf>
 - En litt mer teknisk innføring i generiske typer