IN1010 - Seminar 3

- python → Java
  - I/O

Laget av: Marlen Jarholt (marleja@ifi.uio.no) og Johanna Haarseth (johannph@ifi.uio.no)

Praktisk

Oblig 1 er nå lagt ut, frist 1. februar kl 23.59

Husk å sjekke emnesiden regelmessig

Vise hvor ressurser for hver uke ligger

https://www.uio.no/studier/emner/matnat/ifi/IN1010/v21/index.html

Tips for å komme i gang med oblig 1

Tegn datastrukturtegning

- Hvilke klasser trenger man og hvilke referanser må eksistere?
- Bruk “Programdesign”-delen, mange hint her
- Tegn etterhvert som du leser gjennom oppgaveteksten
- Til slutt: sjekk at dersom du implementerer det du har tegnet, vil systemet da kunne gjøre _alt_ oppgaven ber om?
- Les oppgaveteksten flere ganger

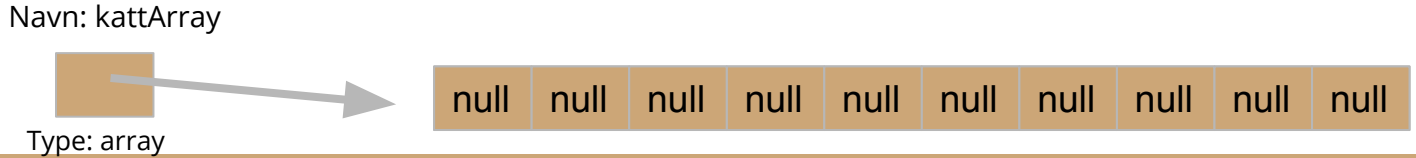
Datastrukturtegningen kan hjelpe deg å få bedre forståelse for systemet, og med bedre forståelse vil du gjør færre feil, og skrive koden raskere.

Repetisjon forrige uke

NullPointer

Viktig å huske at array i java er fixed size, dette er annerledes enn liste i python, vi kan endre innhold, men ikke lengden

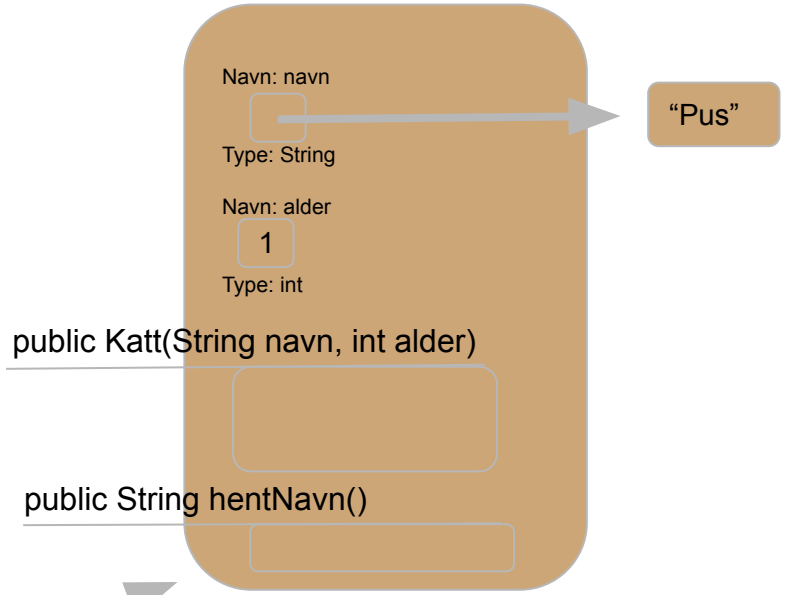
```java
class ArrayEksempel{
  public static void main(String[] args) {
    Katt[] kattArray = new Katt[10];
```

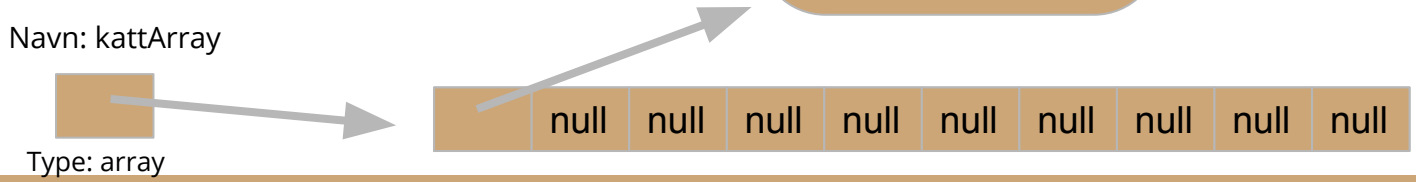


NullPointer

Viktig å huske at array i java er fixed size, dette er annerledes enn liste i python, vi kan endre innhold, men ikke lengden

```
1  class ArrayEksempel{
2    public static void main(String[] args) {
3      Katt[] kattArray = new Katt[10];
4
5      kattArray[0] = new Katt("Pus", 1);
```





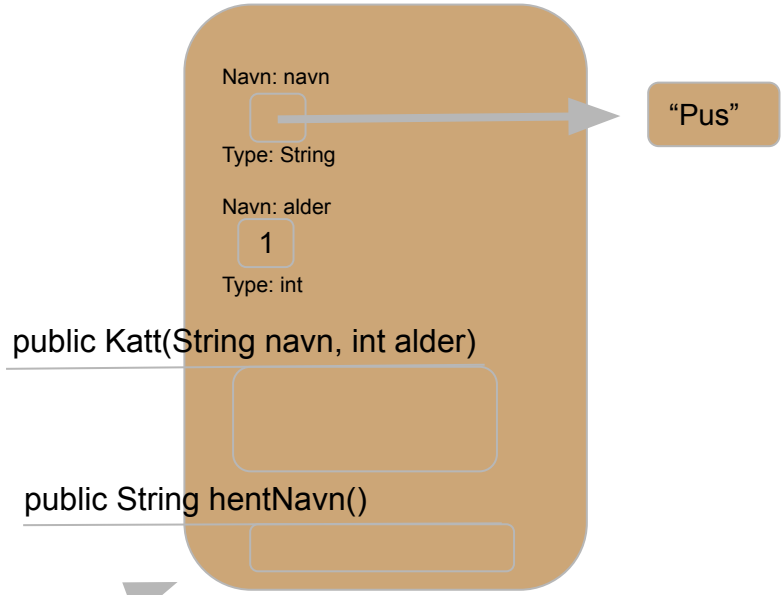
NullPointerException

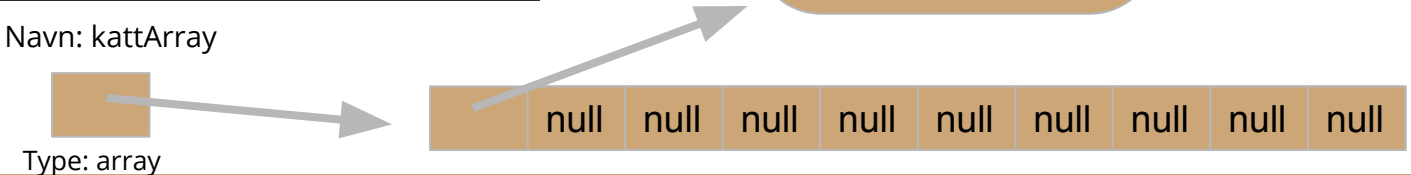
Viktig å huske at array i java er fixed size, dette er annerledes enn liste i python, vi kan endre innhold, men ikke lengden

```
1  class ArrayEksempel{
2    public static void main(String[] args) {
3      Katt[] kattArray = new Katt[10];
4
5      kattArray[0] = new Katt("Pus", 1);
6
7      kattArray[0].hentNavn()
8
9      kattArray[1].hentNavn()
10   }
11 }
```

Evaluerer til "Pus"

null.hentNavn()
Gir error:
java.lang.NullPointerException





Repetisjon denne uken

Arv

Både Fjellrev og Katt er Dyr, det hadde vært bra å kunne kommunisere det på en måte i koden vår!

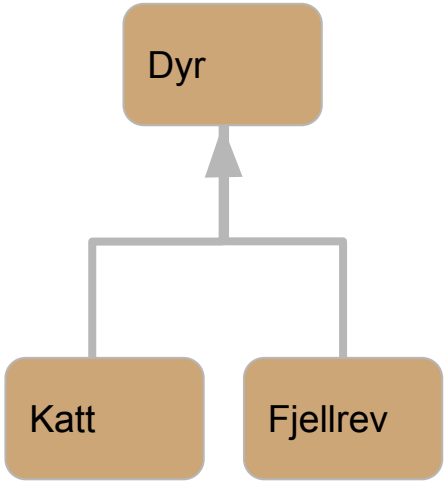
Klassehierarki

Alle Katter er Dyr

Alle Fjellrever er Dyr

Ingen Katter er Fjellrever

Ingen Fjellrever er katter



Klassehierarki

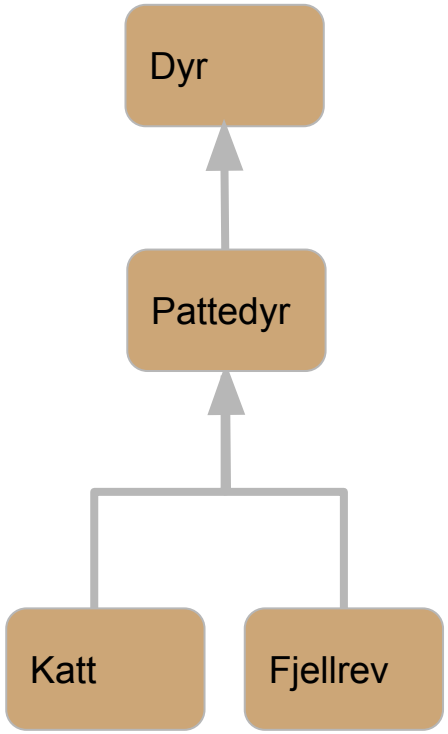
Alle Katter er Dyr

Alle Fjellrever er Dyr

Ingen Katter er Fjellrever

Ingen Fjellrever er katter

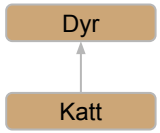
Og man kan ha mange nivåer, f.eks. kan vi legge til pattedyr



Katt og dyr

Her definerer jeg at katter har bostedsadresse, mens andre dyr ikke nødvendigvis har det.

Alle dyr har navn og alder!




```java
class Katt extends Dyr{

  private String bostedsadresse;

  public void settBosted(String bosted){
    bostedsadresse = bosted;
  }
  public String hentBosted(){
    return bostedsadresse;
  }
}
```

```java
class Dyr{
  protected int alder;
  protected String navn;

  public void settAlder(int alder){
    this.alder = alder;
  }
  public  int hentAlder(){
    return alder;
  }
  public  void settNavn(String navn){
    this.navn = navn;
  }
  public  String hentNavn(){
    return navn;
  }
  public void skrivUtInfo(){
    System.out.println("Navn: " + navn);
    System.out.println("Alder: " + alder);
  }
}
```

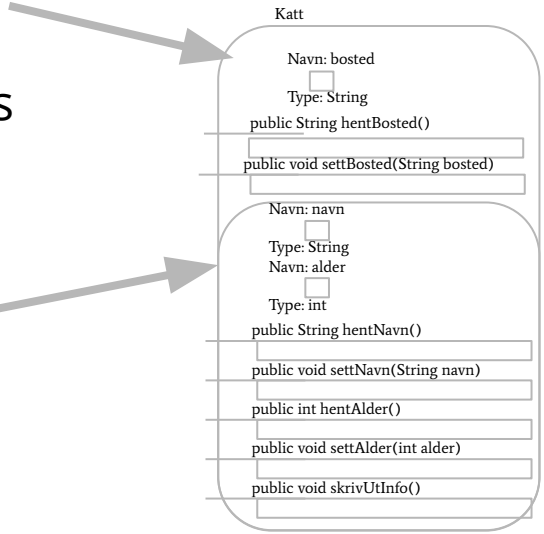
Katt og dyr

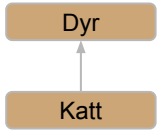
Her definerer jeg at katter har bostedsadresse, mens andre dyr ikke nødvendigvis har det.

Alle dyr har navn og alder!

```java
class Katt extends Dyr{

  private String bostedsadresse;

  public void settBosted(String bosted){
    bostedsadresse = bosted;
  }
  public String hentBosted(){
    return bostedsadresse;
  }
}
```

```java
class Dyr{
  protected int alder;
  protected String navn;

  public void settAlder(int alder){
    this.alder = alder;
  }
  public  int hentAlder(){
    return alder;
  }
  public  void settNavn(String navn){
    this.navn = navn;
  }
  public  String hentNavn(){
    return navn;
  }
  public void skrivUtInfo(){
    System.out.println("Navn: " + navn);
    System.out.println("Alder: " + alder);
  }
}
```





Alle Katter er Dyr

Alle katter får automatisk alle egenskapene til Dyr!

Husk at metoden settNavn() og HentNavn() var metoder i klassen Dyr, og ikke metoder i klassen Katt, siden alle Dyr har navn (i dette programmet).

Det går helt fint å legge en referanse til et objekt av klassen Katt i variabelen katt2, fordi Katt er et Dyr

Arv

En referanse til et Katteobjekt kan ligge i en variabel med type Dyr og Katt

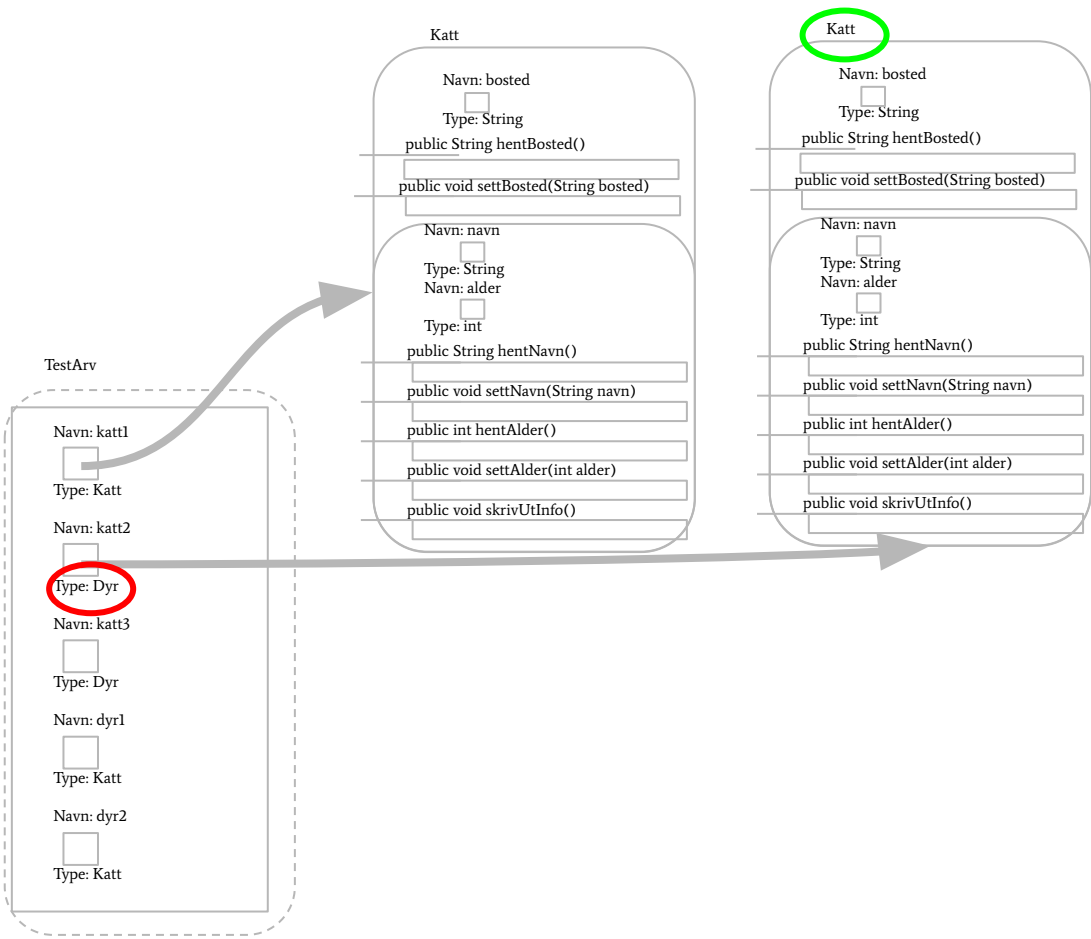
En referanse til et Dyreobjekt kan **ikke** ligge i en variabel med type Katt, heller ikke hvis vi caster.

Generelt: god stil å unngå casting så mye som du kan!

```
1  class TestArv{
2    public static void main(String[] args) {
3      // Lov!
4      Katt katt1 = new Katt();
5      Dyr katt2 = new Katt();
6      // Lov
7      Katt katt3 = (Katt) katt2;
8      // Ikke lov
9      Katt Dyr1 = new Dyr();   Error
10     // heller ikke lov
11     Katt Dyr2 = (Katt) new Dyr();   Error
12
```

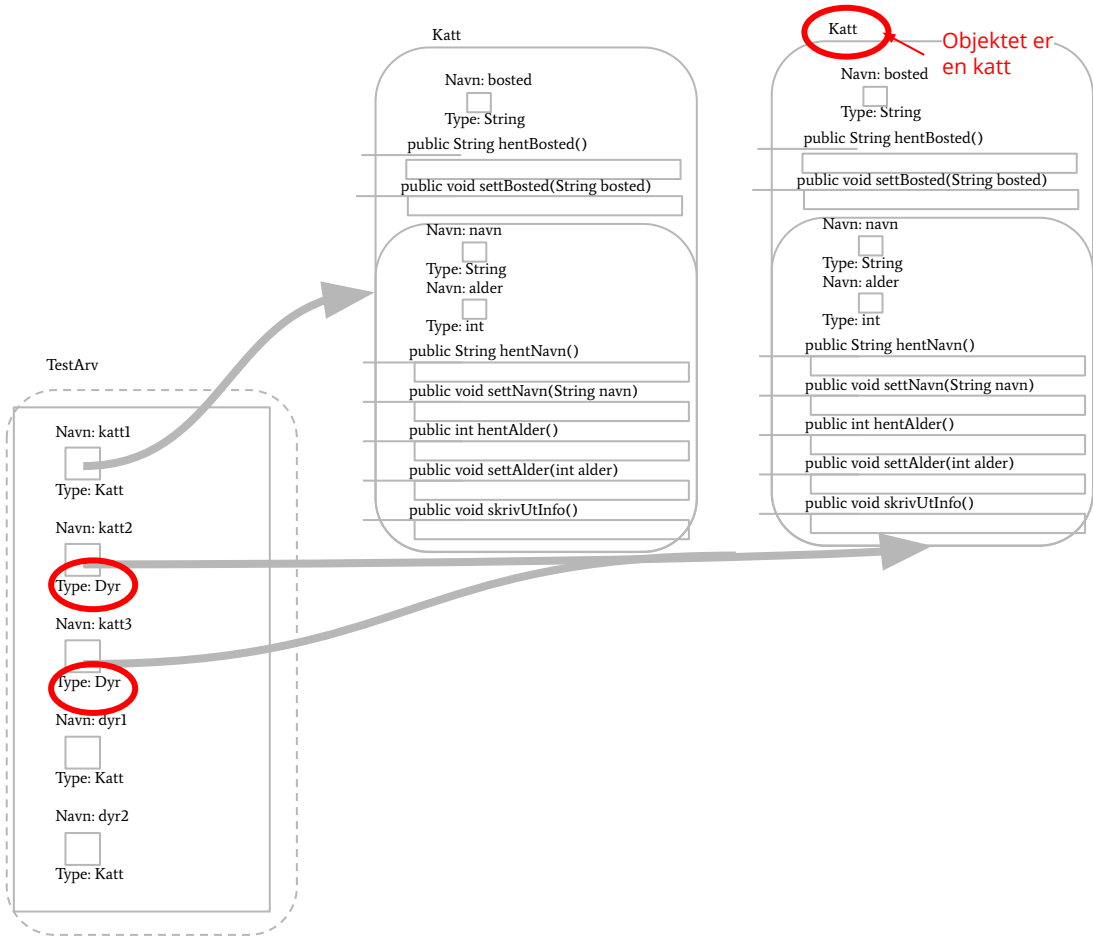
Datastrukturtegning

```java
class TestArv{
  public static void main(String[] args) {
    // Lov!
    Katt katt1 = new Katt();
    Dyr katt2 = new Katt();
    // Lov
    Katt katt3 = (Katt) katt2;
    // Ikke lov
    Katt dyr1 = new Dyr();    Error
    // heller ikke lov
    Katt dyr2 = (Katt) new Dyr(); Error

```



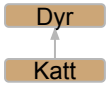
Datastrukturtegning

```
1  class TestArv{
2    public static void main(String[] args) {
3      // Lov!
4      Katt katt1 = new Katt();
5      Dyr katt2 = new Katt();
6      // Lov
7      Katt katt3 = (Katt) katt2;
8      // Ikke lov
9      Katt dyr1 = new Dyr();     Error
10     // heller ikke lov
11     Katt dyr2 = (Katt) new Dyr(); Error
12
```



Casting betyr at vi forteller kompilatoren at vi vet hva vi driver med! Vi vet at objektet som katt2 referer til er en Katt. Hvis vi ikke caster får vi kompileringsfeil

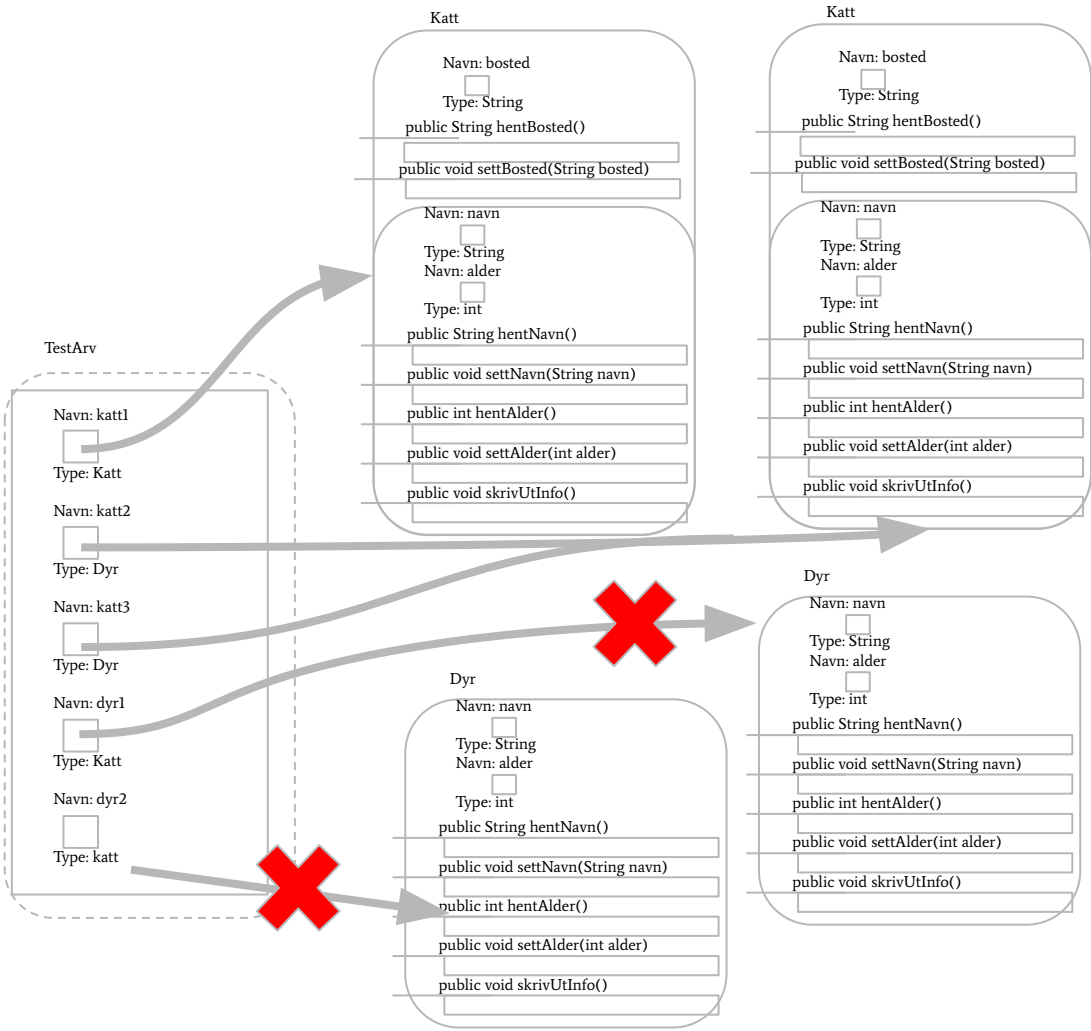
Må caste her fordi typen til variablen katt2 er Dyr.



Datastrukturtegning

```
1  class TestArv{
2    public static void main(String[] args) {
3      // Lov!
4      Katt katt1 = new Katt();
5      Dyr katt2 = new Katt();
6      // Lov
7      Katt katt3 = (Katt) katt2;
8      // Ikke lov
9      Katt dyr1 = new Dyr();     Error
10     // heller ikke lov
11     Katt dyr2 = (Katt) new Dyr(); Error
12
```

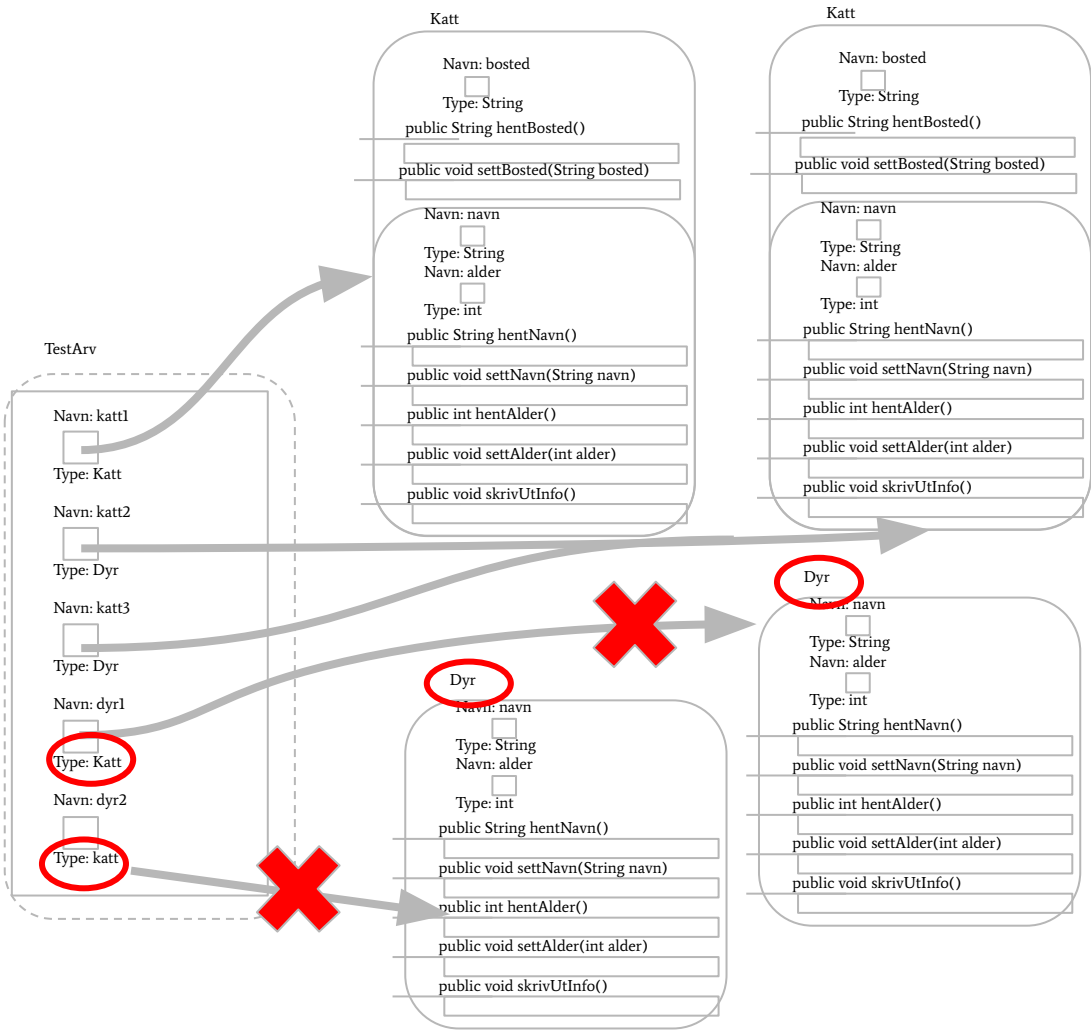
1. Vi får kompileringsfeil
   - ➢ "Løses" med casting, kompilatoren klager ikke fordi vi har sagt at vi vet hva vi driver med
2. Dyr er ikke Katt: run time error
   - ➢ kan ikke løses



Datastrukturtegning

```java
class TestArv{
  public static void main(String[] args) {
    // Lov!
    Katt katt1 = new Katt();
    Dyr katt2 = new Katt();
    // Lov
    Katt katt3 = (Katt) katt2;
    // Ikke lov
    Katt dyr1 = new Dyr();    Error
    // heller ikke lov
    Katt dyr2 = (Katt) new Dyr(); Error

```

1. Vi får kompileringsfeil
   - ➢ "Løses" med casting, kompilatoren klager ikke fordi vi har sagt at vi vet hva vi driver med
2. Dyr er ikke Katt: run time error
   - ➢ kan ikke løses



Arv

En referanse til et Katteobjekt kan ligge i en variabel med type Dyr og Katt

En referanse til et Dyreobjekt kan **ikke** ligge i en variabel med type Katt, heller ikke hvis vi caster.

Generelt: god stil å unngå casting så mye som du kan!

```
1  class TestArv{
2    public static void main(String[] args) {
3      // Lov!
4      Katt katt1 = new Katt();
5      Dyr katt2 = new Katt();
6      // Lov
7      Katt katt3 = (Katt) katt2;
8      // Ikke lov
9      Katt Dyr1 = new Dyr();   Error
10     // heller ikke lov
11     Katt Dyr2 = (Katt) new Dyr();   Error
12
```

Error-håndtering

Her ser dere to eksempler på kode som gjør det samme(åpner en fil)

1) Med error håndtering (try/catch)
2) Uten error-håndtering, sender problemet viderer (throws).

   Da trenger man try/catch når man kaller på denne metoden ellers vil programmet terminere(avslutte). (Siden dette er main som "Throws exception" vil programmet terminere ved error.)

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

// Med error-håndtering
class FraFil1{
  public static void main(String[] args){
    // Åpne datafilen:
    Scanner fil = null;
    try {
      fil = new Scanner(new File(filNavn));
    } catch (Exception e) {
        System.out.println("Kan ikke lese " + filNavn + "!");
        System.exit(1);
    }
  }
}

// Uten error-håndtering
class FraFil2{
    public static void main(String[] args) throws Exception {
        Scanner fil = new Scanner(new File("tekst.txt"));
        }
    }
}
```

Med errorhåndtering

```
jonbon@jons-macbook-pro uke2 % java FraFil1
Kan ikke lese filen !
jonbon@jons-macbook-pro uke2 % java FraFil2
Exception in thread "main" java.io.FileNotFoundException: tekst.txt (No such file or directory)
        at java.base/java.io.FileInputStream.open0(Native Method)
        at java.base/java.io.FileInputStream.open(FileInputStream.java:219)
        at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
        at java.base/java.util.Scanner.<init>(Scanner.java:639)
        at FraFil2.main(Errorhåndtering.java:21)
```

Uten errorhåndtering

Send Johanna en direktemelding i chatten

1. Vil du jobbe med andre i breakoutrooms?
2. Hvor godt forstår du stoffet fra denne uken på en skala fra 1(lite godt) - 6(veldig godt)?
3. Noen spesielle du vil jobbe med?

Pause 15min

Jobbe med oppgaver

Breakoutrooms

1. Slå på kamera og ha en presentasjonsrunde
2. Diskuter:
   a. Hva betyr det at en variabel med type Dyr kan ha en referanse til et objekt av klassen Katt, men ikke omvendt?
      (Variabel av type A kan referere til objekt av B, men ikke omvendt)
   b. Hva betyr kodeordet Protected? (Google gjerne eller se i forelesning hvis dere ikke husker!)
      Hva er forskjellen på private, public og protected?
   c. Hva gjør metoden instanceOf()?
3. Jobb sammen med ukesoppgavene, de ligger på emnesiden -> grupper
   a. Enten ved at én deler skjerm eller med jdoodle.com eller med codecollab.io/
   b. OBS: både jdoodle og codecollab er gratistjeneseter som UiO ikke har avtale med, sannsynligvis vil de samle data om dere. Dere kan fint løse oppgavene uten å bruke disse tjenestene!
4. Bruk “ask for help”-knappen for å få hjelp 😊
5. Vi møtes her igjen for å gå gjennom oppgavene til slutt (dere bestemmer hvilke)

Ris, ros, forslag ?

https://nettskjema.no/a/180345

Tegning oppgave 4