Repetisjon

GRENSESNITT, REFERANSER OG TYPEKONVERTERING

MIAU@IFI.UIO.NO

Plan for dagen

- Interface
- Referanser
- Eksplisitt typekonvertering
- Implisitt typekonvertering
- Referanser til interface

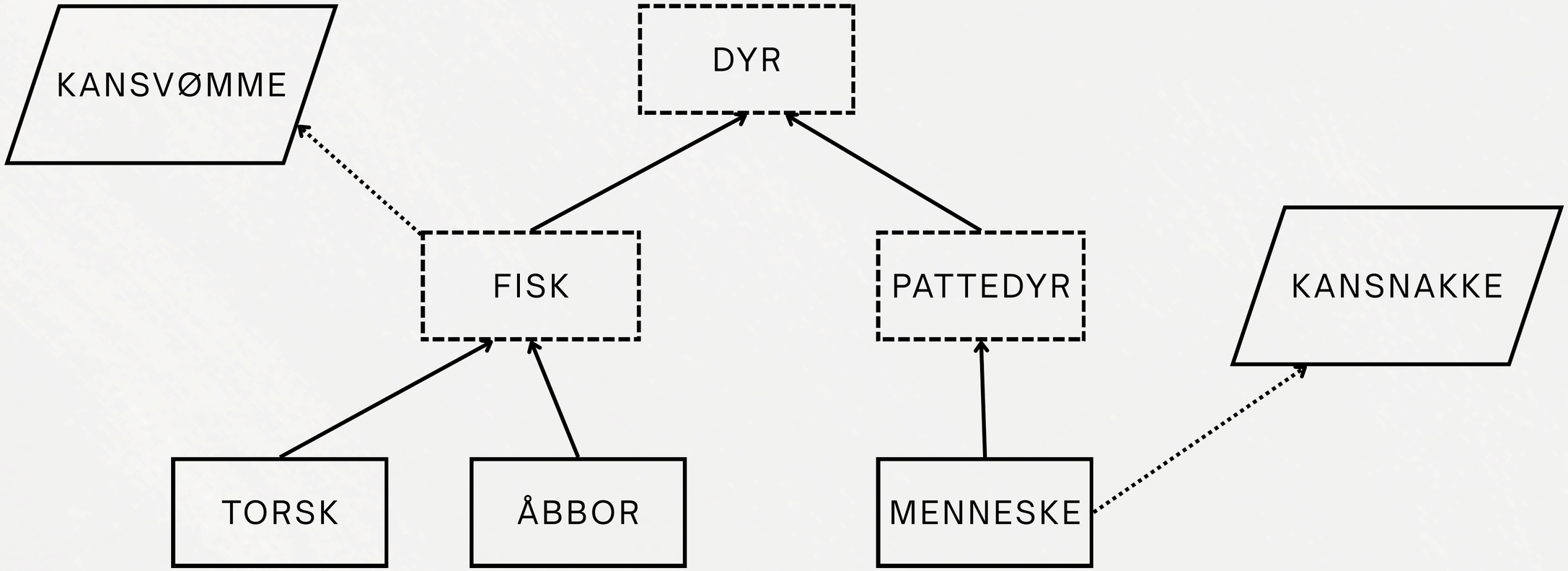
Interface

- Metodefinisjoner, uten implementasjon
  - Klasser kan implementere interface, og må dermed implementere alle metodene i interfacet
- Kan ikke opprette instanser av et interface
- Kan opprette referansevariabler med interface som type
- Nyttig for å kategorisere klasser etter egenskaper
  - To klasser med like egenskaper, hvor arv ikke vil gi mening

```java
interface LeggerEgg {
    void leggEgg();
    void antallEgg();
}
```

```java
abstract class Fisk implements LeggerEgg {
    @Override
    public void leggEgg() {
        System.out.println(x: "Fisken legger egg.");
    }

    @Override
    public void antallEgg() {
        System.out.println(x: "Fisken la 600 egg");
    }
}
```


KANSVØMME
DYR
FISK
PATTEDYR
KANSNAKKE
TORSK
ÅBBOR
MENNESKE

Referanser

- Variabler som peker på objekter
- Har en type som bestemmer hvilke objekter de kan peke på
  - Kan peke på instanser av alle klasser som arver fra typen (implisitt typekonvertering)

```
Fisk fisk = new Laks();
```

- Kun tilgang til metodene til deres egen type
  - Selv hvis den peker på et objekt som har flere metoder
- ‘fisk’ vil ikke ha tilgang til en unik metode i Laks

Implisitt typekonvertering

- En peker har en type
  - Kan peke på objekter som er deres egen type, eller av en subklasses type
- Kun tilgang til pekeren sin type sine egenskaper

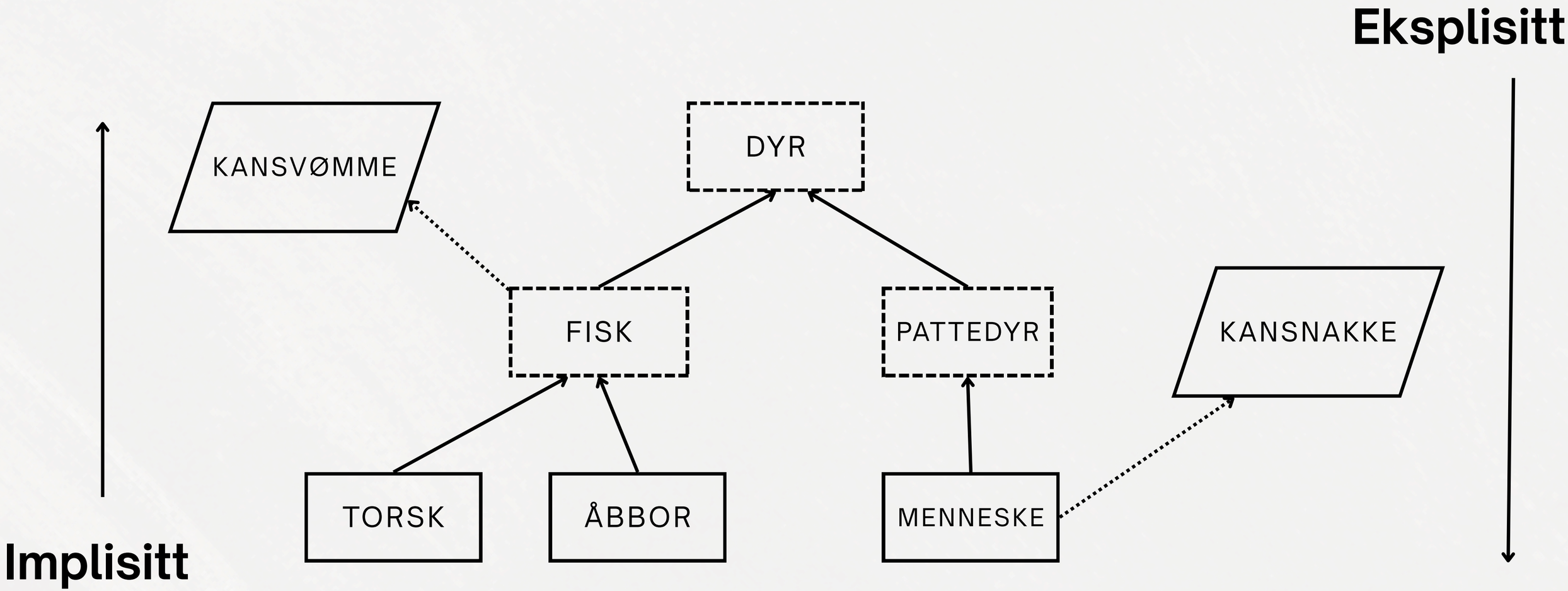
Eksplisitt typekonvertering

- For å få tilbake egenskapene i subklassen, må man eksplitt typekonverte objektet

```
if (fisk instanceof Laks) {
    Laks laks = (Laks) fisk;
}
```

- Må lage en ny referansevariabel ved eksplisitt konvertering
- Man kan kun typekonvertere et objekt til en type som objektet er en instans av (alle klasser/interface som den arver fra/implementerer)

Konvertering



Referanser til interface

- Kan ha referansevariabler til interface
- Kan kun peke på objekter som implementerer interfacet
  - Referansevariabelen vil kun ha tilgang til metodene i interfacet
- Nyttig for å kategorisere objekter med de samme egenskapene

```
LeggerEgg laks = new Laks();
```

- Må bruke eksplisitt konvertering for å typekonvertere tilbake