IN1010 uke 4

Gruppe 9

Agenda

- Repetisjon fra uke 3 forelesning:
  Arv og subklasser → polymorfi
  - @Override
  - *Object* - toString(), equals()
  - Super()
  - Overloading
- Intro til denne ukens stoff:
  Interface (grensesnitt)
- Jobbe med oppgaver/trix/oblig!

Polymorfi

Repetisjon: Hunde-eksempelet

```java
class Hund extends Dyr {
    public void bjeff() {
        System.out.println("Voff");
    }
}
```

```java
class GoldenRetriever extends Hund {
    public void vaerGlad() {
        System.out.println("Jeg er såå glad!");
    }
}
```

Hovedprogram:

```java
public static void main(String[] args) {
    Hund minHund = new GoldenRetriever();
    minHund.bjeff();
    minHund.vaerGlad();
}
```

```java
class Hund extends Dyr {
    public void bjeff() {
        System.out.println("Voff");
    }
}
```

```java
class GoldenRetriever extends Hund {
    public void bjeff() {
        System.out.println("Glad voff");
    }
}
```

Hovedprogram:

```java
public static void main(String[] args) {
    Hund minHund = new Hund();
    Hund minRetriever = new GoldenRetriever();
    GoldenRetriever minRetriever2 = new GoldenRetriever();

    minHund.bjeff();
    minRetriever.bjeff();
    minRetriever2.bjeff();
}
```

Vil programmet kjøre?
Hva skjer?

```java
class Hund extends Dyr {
    public void bjeff() {
        System.out.println("Voff");
    }
}
```

```java
class GoldenRetriever extends Hund {
    public void bjeff() {
        System.out.println("Glad voff");
    }
}
```

Hovedprogram:

```java
public static void main(String[] args) {
    Hund minHund = new Hund();
    Hund minRetriever = new GoldenRetriever();
    GoldenRetriever minRetriever2 = new GoldenRetriever();

    minHund.bjeff();
    minRetriever.bjeff();
    minRetriever2.bjeff();
}
```

Hvorfor?
Polymorfi!

```
Voff
Glad voff
Glad voff
```

```java
class Hund extends Dyr {
    public void bjeff() {
        System.out.println("Voff");
    }
}
```

```java
class GoldenRetriever extends Hund {
    @Override
    public void bjeff() {
        System.out.println("Glad voff");
    }
}
```

- **Overskriver** metoden i *Hund*
  → Vil alltid bruke den metoden i subklassen "lengst ned"

- @Override → for å forsikre at vi faktisk overskriver en metode som finnes i superklassen

```java
class Hund extends Dyr {
    public void bjeff() {
        System.out.println("Voff");
    }
}
```

```java
class GoldenRetriever extends Hund {
    public void bjff() {
        System.out.println("Glad voff");
    }
}
```

- **Overskriver** metoden i *Hund*
  → Vil alltid bruke den metoden i subklassen "lengst ned"

- @Override → for å forsikre at vi faktisk overskriver en metode som finnes i superklassen

```java
class Hund extends Dyr {
    public void bjeff() {
        System.out.println("Voff");
    }
}
```

- **Overskriver** metoden i *Hund*
  → Vil alltid bruke den metoden i subklassen "lengst ned"

```java
class GoldenRetriever extends Hund {
    @Override
    public void bjff() {
        System.out.println("Glad voff");
    }
```

⊗ The method bjff() of type GoldenRetriever must override or implement a supertype method

- @Override → for å forsikre at vi faktisk overskriver en metode som finnes i superklassen

Kodeeksempel: Figurer

- Polymorfi

@Override toString(), equals() fra Object

- Overloading
- Super()

Interface

(Intro til forelesningen som er senere i dag)

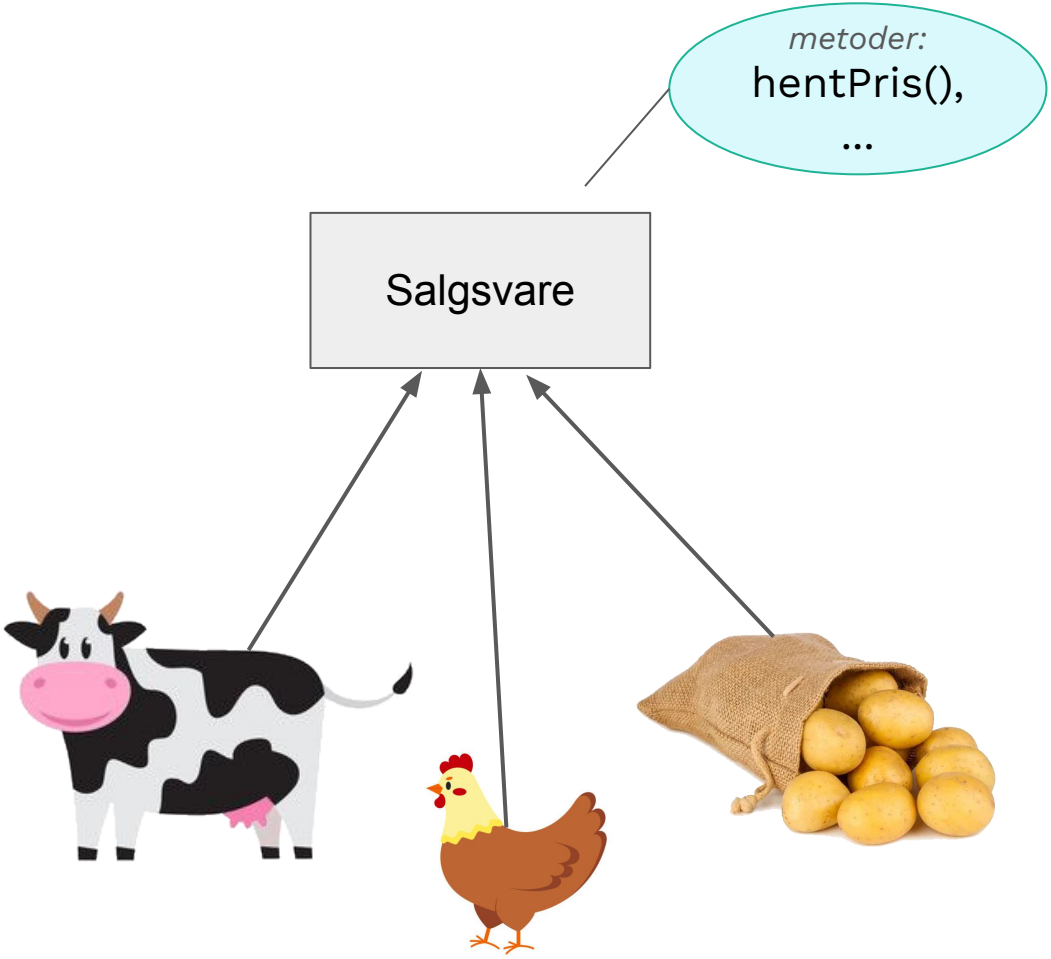
Interface

- Hva/hvorfor?
- Forskjellen på interface/abstrakt klasse/klasse?

Hvorfor interface?

Eksempel: Bondegårdsystem

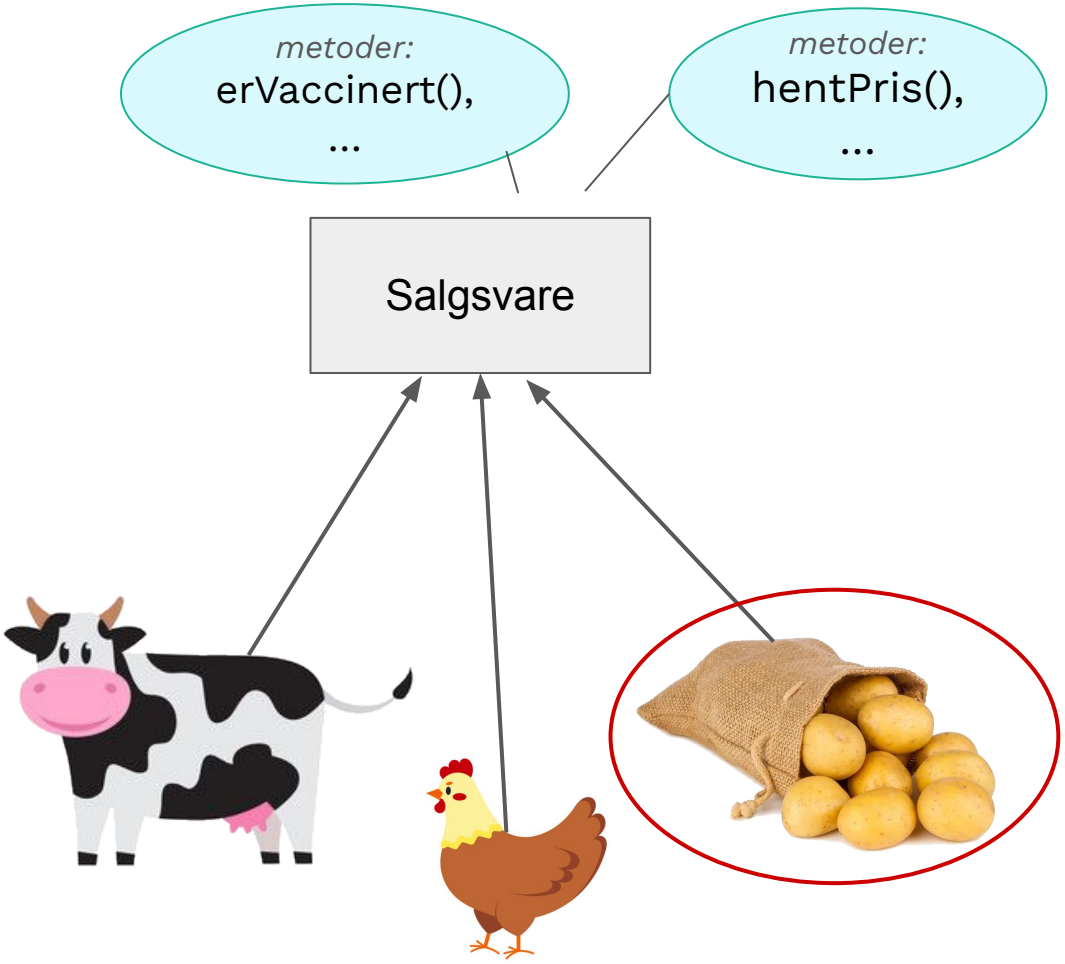
- Både selge dyr og sjekke helsen



Hvorfor interface?

Eksempel: Bondegårdsystem

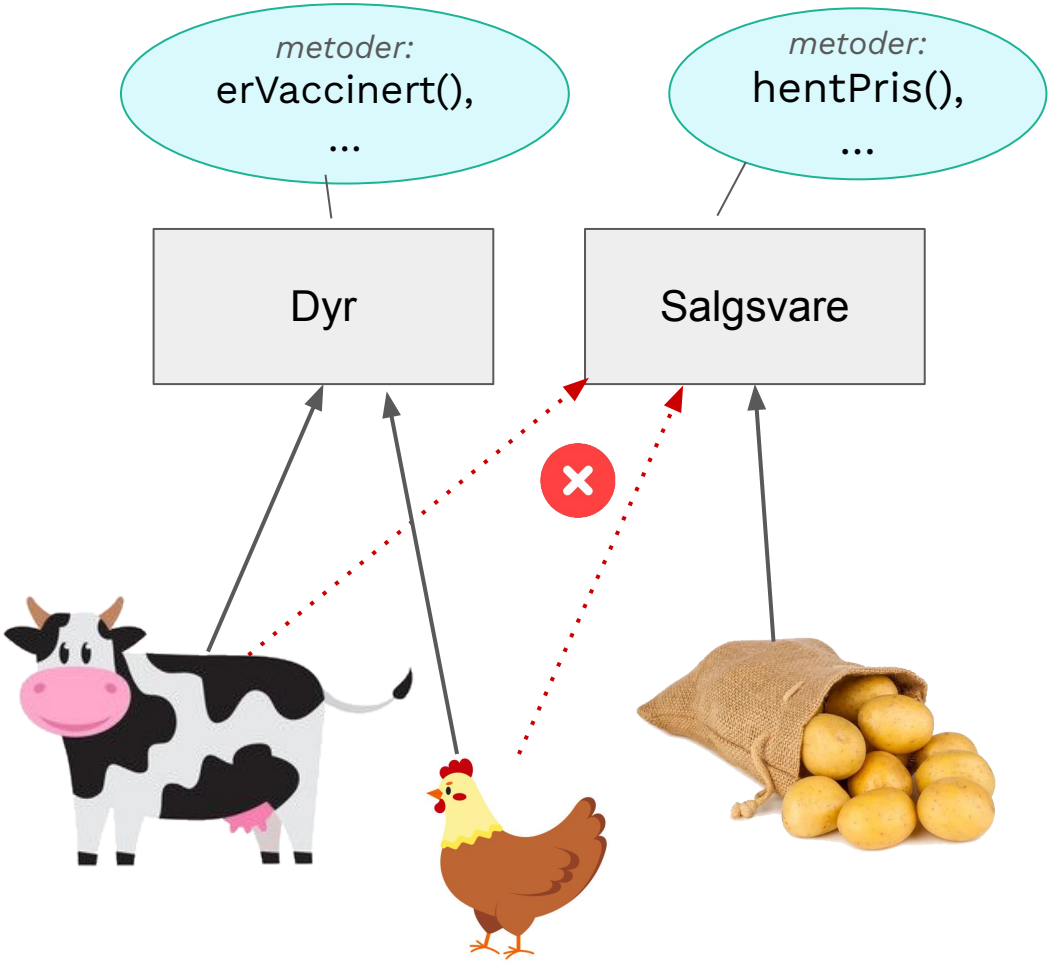
- Både selge dyr og sjekke helsen
  → legg til dette i superklasse??



Hvorfor interface?

Eksempel: Bondegårdsystem

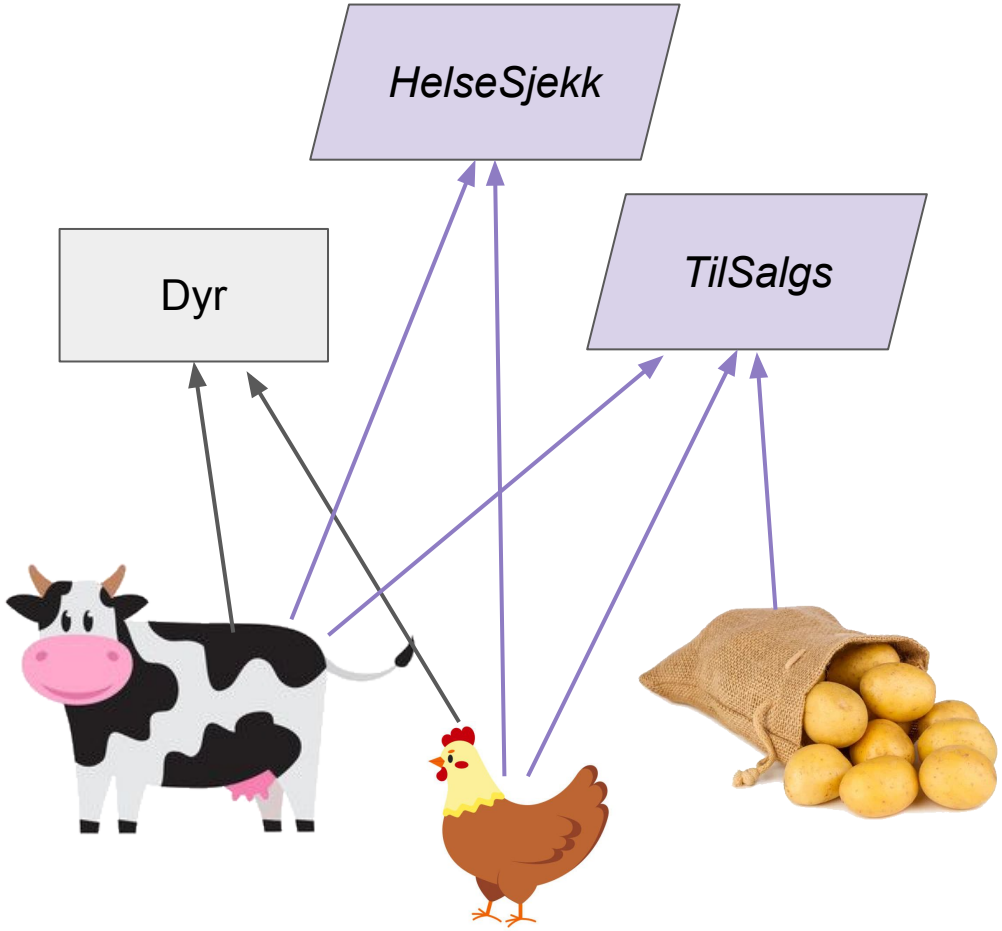
- Både selge dyr og sjekke helsen
  → Arve fra en til klasse?
  - kan bare arve fra én



Hvorfor interface?

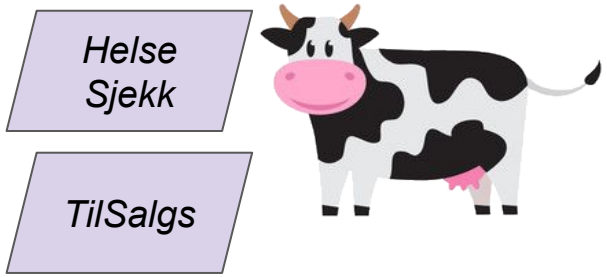
Eksempel: Bondegårdsystem

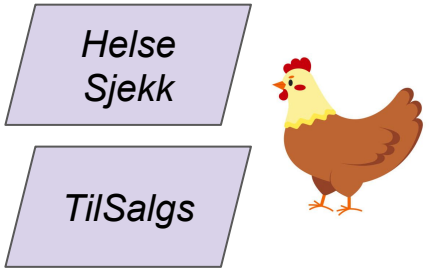
- Både selge dyr og sjekke helsen
  → løsning: interface



Hvorfor interface?

- En klasse kan implementere flere interface
- Spesifiserer rollen/grensesnittet til klassen
  - “Utenifra”: Vet hva klassen gjør uten å måtte se implementasjonen
  - Programmereren: Vet hva klassen må kunne gjøre men har frihet til å implementere det (skrive koden) som man vil







```java
class Poteter implements TilSalgs{
    private double vekt;
    private String dato;
    private int kgPris = 8;

    public Poteter(double vekt, String dato) {
        this.vekt = vekt;
        this.dato = dato;
    }

    @Override
    public int hentPris() {
        return (int) Math.round(vekt * kgPris) ;
    }

    @Override
    public void settPris(int nyPris) {
        kgPris = nyPris;
    }
```

```java
interface TilSalgs {
    int hentPris();
    void settPris(int nyPris);
}
```



```java
class Ku extends Dyr implements TilSalgs, HelseSjekk {
    private String navn;
    private double vekt;
    private int pris = 11000;
    private boolean vaksinert = false;

    public Ku(String navn, int alder, double vekt) {
        super(alder);
        this.navn = navn;
        this.vekt = vekt;
    }

    @Override
    public int hentPris() {
        return pris;
    }

    @Override
    public void settPris(int nyPris) {
        pris = nyPris;
    }
```

```java
interface TilSalgs {
    int hentPris();
    void settPris(int nyPris);
}
```

- Ulik implementering!



```java
@Override
public boolean normalVekt() {
    return vekt > 500 && vekt < 750;
}

@Override
public boolean erVaksinert() {
   return vaksinert;
}

@Override
public void giVaksine() {
    if (!erVaksinert()) {
        vaksinert = true;
        System.out.println("Vaksinerte kuen" + navn);
    }
}
```

```java
interface HelseSjekk {
    boolean normalVekt();
    boolean erVaksinert();
    void giVaksine();
}
```



```java
class BondegaardSystem {
    Run | Debug
    public static void main(String[] args) {
        Poteter potet1 = new Poteter(2.5, "08.08.2022");
        Poteter potet2 = new Poteter(2.3, "08.08.2022");
        Ku ku = new Ku("Molly", 3, 550);

        TilSalgs[] salgListe = {potet1, potet2, ku};

        for (TilSalgs vare : salgListe) {
            System.out.println(vare + ": " + vare.hentPris() + "kr");
        }
    }
}
```

```
Poteter 2.5 kg: 20kr
Poteter 2.3 kg: 18kr
Kuen Molly, 3 aar: 11000kr
```

Vi kan gjøre dette fordi vi har peker av type TilSalgs!

Jobb med oppgaver!

IN1010 Emnesiden → Grupper →
Gruppe 9 → Uke4

Husk å ta pause ☕