IN1010 uke 3

Gruppe 9

Kort menti

Gå inn på *mentimeter.com*

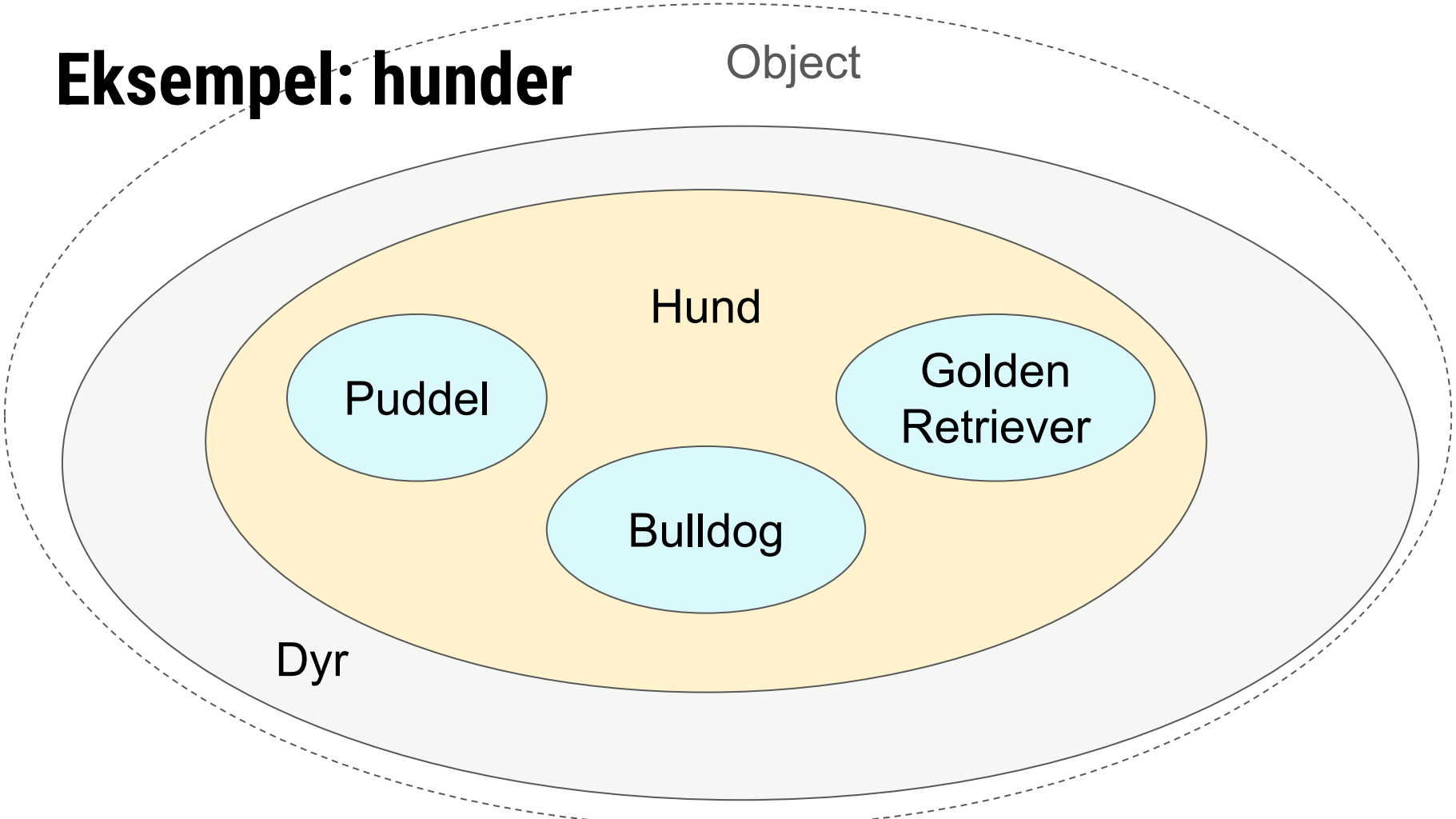
Agenda

- Repetisjon av arv og subklasser
  - Arv vs komposisjon
  - Referansevariabler
  - Casting
  - *instanceOf*
  - *private*/*protected*/*public* variabler
  - Abstrakte klasser
- Jobbe med oppgaver/trix/oblig!

Arv og subklasser

1 minutt: skriv ned alt du husker om arv og subklasser!

Eksempel: hunder



Eksempel: hunder

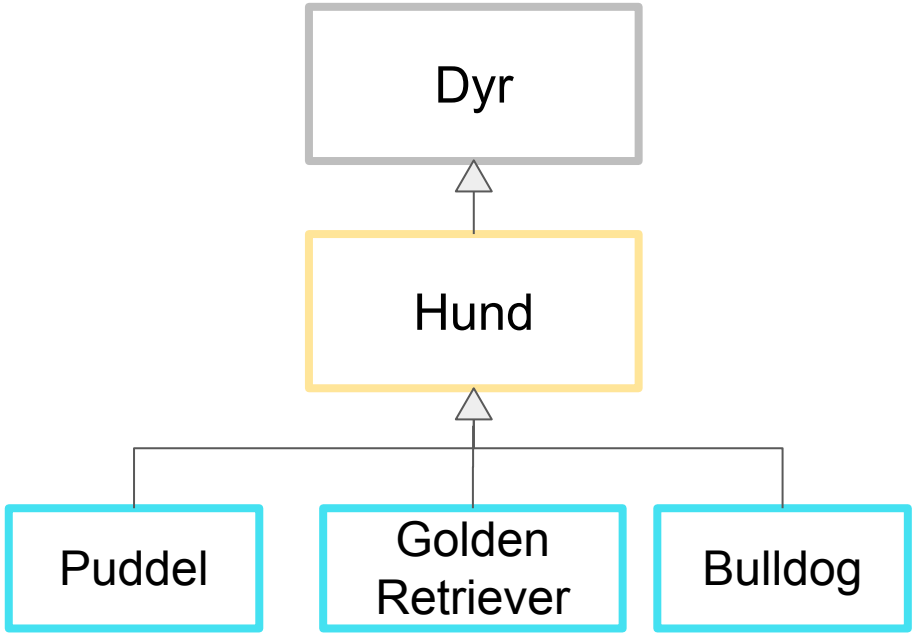
```
class Dyr

class Hund extends Dyr

class Puddel extends Hund

class Bulldog extends Hund

class GoldenRetriever extends Hund
```



Arv vs komposisjon Likheter

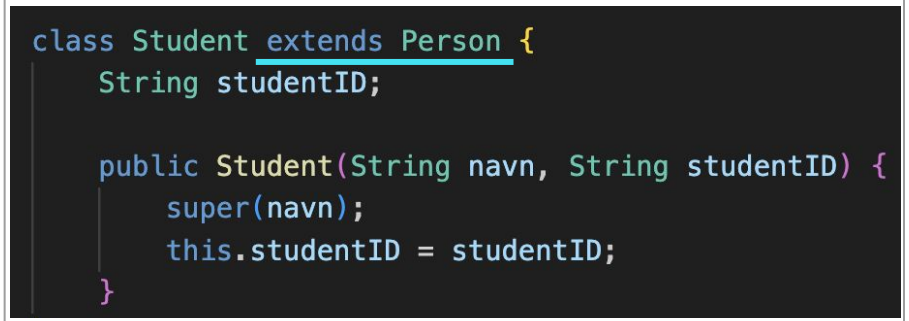
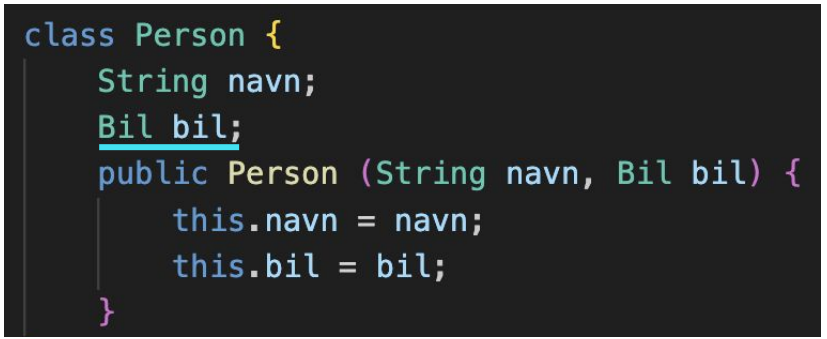
Begge...

- Er fundamentale konsepter innen objektorientert programmering (OOP)
- Brukes til å gjenbruke og strukturere kode

Arv vs komposisjon Forskjeller

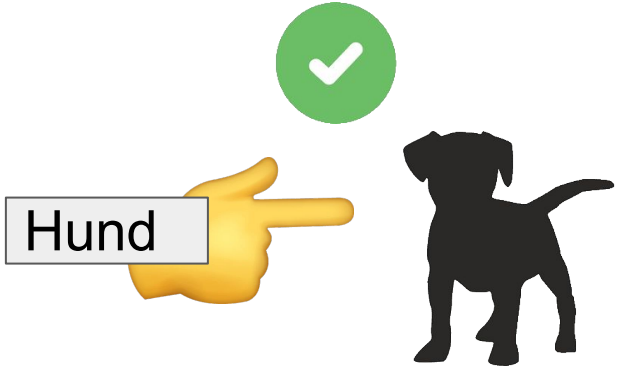
Arv	Komposisjon
• *"er-en"*-forhold "En Student er en Person" • *Subklassen* arver metoder og instansvariabler fra *superklassen* • En klasse kan bare arve fra **én** annen klasse	• *"har-en"*-forhold "En Person har en Bil" "Et Rutenett har flere Celler" • Klasser bruker andre klasser ved å ha instansvariabler som peker på objekt av de andre klassene • En klasse kan bygges opp av **flere** andre klasser

Arv vs komposisjon Forskjeller

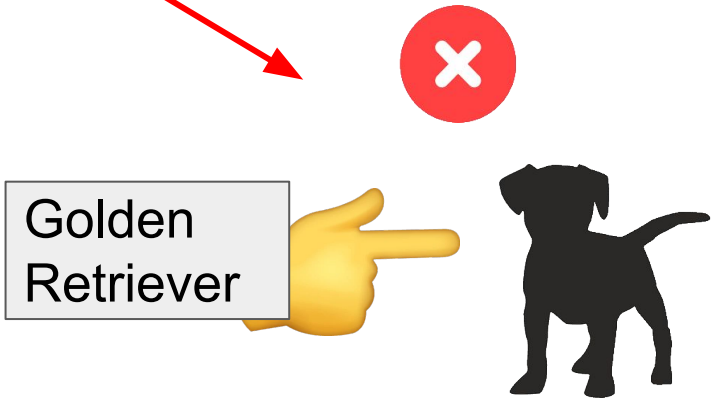
Arv	Komposisjon
	
“En Student er en Person”	“En Person har en Bil”

Referansevariabler

```
Hund hund1 = new Hund();
Hund hund2 = new GoldenRetriever();
GoldenRetriever golden1 = new Hund();
```







```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!");
    }
}
```

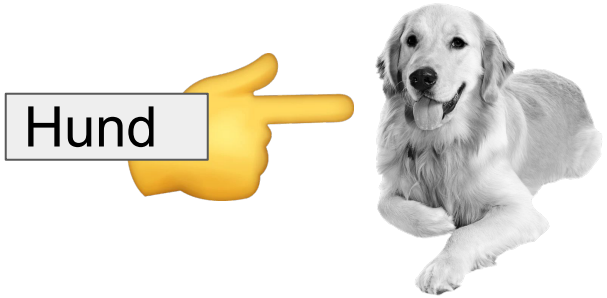
Hovedprogram:

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

Hva skjer når vi kjører hovedprogrammet?

⊗ The method vaerGlad() is undefined for the type Hund

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



- Pekeren/variabelen “ser” bare at det er en Hund, ikke GoldenRetriever-egenskapene

Casting

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

    GoldenRetriever minRetriever = (GoldenRetriever) minHund;
    minRetriever.bjeff();
    minRetriever.vaerGlad();
}
```



InstanceOf

- Sjekker om et objekt er av, eller arver fra, en klasse

Hva printes ut?

```
Hund minHund = new GoldenRetriever();
Hund minHund2 = new Bulldog();
GoldenRetriever minRetriever = (GoldenRetriever) minHund;

System.out.println(minHund instanceof GoldenRetriever);
System.out.println(minHund2 instanceof GoldenRetriever);
System.out.println(minRetriever instanceof GoldenRetriever);
System.out.println(minRetriever instanceof Hund);
```

```
true
false
true
true
```

Protected

- private - bare synlig i klassen
- protected - synlig i klassen og alle subklasser
- public - synlig i alle klasser

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

Abstrakte klasser

- Kan ikke lage objekt av dem
- Men kan arve fra dem

```
abstract class Dyr {
    private boolean sulten = true;
    public abstract void spis();
}
```

Jobb med oppgaver/oblig!

IN1010 Emnesiden → Grupper →
Gruppe 9 → Uke3

Husk å ta pause ☕