IN1010 uke 3

Gruppe 2

Agenda

- Arv vs komposisjon
- Arv og subklasser → polymorfi
  - @Override
  - *Object* - toString(), equals()
  - Super()
  - Overloading
- Jobbe med oppgaver/trix/oblig!

Arv vs komposisjon

Arv vs komposisjon Likheter

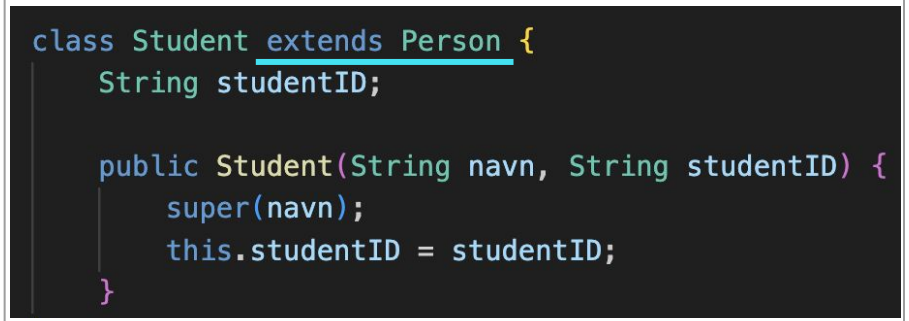
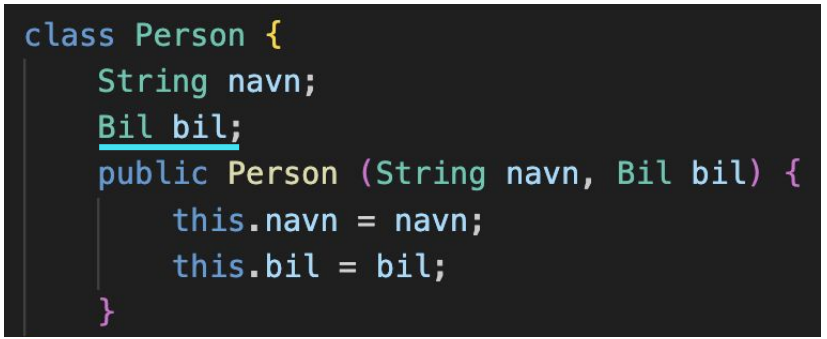
Begge...

- 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”

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()

Jobb med oppgaver!

IN1010 Emnesiden → Grupper →
Gruppe 2 → Uke3