



IN1010 uke 4

Gruppe 4

Agenda

- Arv og subklasser → polymorfi
 - @Override
 - *Object* - toString(), equals()
 - Super()
 - Overloading
- Mer om exceptions
- Litt om oblig 2
- Jobbe med oppgaver/trix/oblig!

Polymorfi



Repetisjon: Hunde-eksempelet

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

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

Hovedprogram:

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

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

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

Hovedprogram:

```
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?

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

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

Hovedprogram:

```
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
```

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

```
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

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

```
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


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

```
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

- **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

Polymorfi kodeeksempel



Mer om exceptions



```
public static void main(String[] args) {
    Scanner fil = null;
    int[] tall = new int[3];
    try {
        fil = new Scanner(new File("minfil.txt"));
        tall[4] = 1;
    } catch (Exception e) {
        System.out.println("Kunne ikke lese filen!");
        System.exit(1);
    }

    while (fil.hasNextLine()) {
        System.out.println(fil.nextLine());
    }
}
```

☰ minfil.txt

```
1   linje 1
2   linje 2
3   linje 3
```

Hva skjer når
vi kjører
programmet?

```
public static void main(String[] args) {
    Scanner fil = null;
    int[] tall = new int[3];
    try {
        fil = new Scanner(new File("minfil.txt"));
        tall[4] = 1;
    } catch (Exception e) {
        System.out.println("Kunne ikke lese filen!");
        System.exit(1);
    }

    while (fil.hasNextLine()) {
        System.out.println(fil.nextLine());
    }
}
```

☰ minfil.txt

```
1   linje 1
2   linje 2
3   linje 3
```

Hva skjer når
vi kjører
programmet?

Kunne ikke lese filen!

```
import java.io.FileNotFoundException;
```

```
public static void main(String[] args) {  
    Scanner fil = null;  
    int[] tall = new int[3];  
    try {  
        fil = new Scanner(new File("minfil.txt"));  
        tall[4] = 1;  
    } catch (FileNotFoundException e) {  
        System.out.println("Kunne ikke lese filen!");  
        System.exit(1);  
    }  
  
    while (fil.hasNextLine()) {  
        System.out.println(fil.nextLine());  
    }  
}
```

≡ minfil.txt

1 linje 1

2 linje 2

3 linje 3

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException:
Index 4 out of bounds for length 3

```
public static void main(String[] args) {
    Scanner fil = null;
    int[] tall = new int[3];
    try {
        fil = new Scanner(new File("minfil.txt"));
        tall[4] = 1;
    } catch (FileNotFoundException e) {
        System.out.println("Kunne ikke lese filen!");
        System.exit(1);
    } catch (IndexOutOfBoundsException e) {
        System.out.println("Kunne ikke sette inn i tall-array!");
        System.exit(1);
    }

    while (fil.hasNextLine()) {
        System.out.println(fil.nextLine());
    }
}
```

☰ minfil.txt

```
1   linje 1
2   linje 2
3   linje 3
```

Altså: Ved try/catch, catch helst en spesifikk exception og ikke alle (*Exception*)

Kunne ikke sette inn i tall-array!

Oblig 2



Jobb med oppgaver!

IN1010 Emnesiden → Grupper →
Gruppe 4 → Uke4

