Velkommen!

...

Johanna
johannph på mattermost
johannph@uio.no på mail!

Kort: Praktisk informasjon

- Undervisningstilbud
  - https://www.uio.no/studier/emner/matnat/ifi/IN1010/v21/undervisningstilbud/
  - Jeg har konkrete spørsmål/problemer med min kode -> Labtime!
  - Jeg vil ha mer liveprogrammering -> Plenumstime!
  - Jeg vil jobbe med andre (og kanskje en kjapp recap av forelesning) -> Gruppetime!
  - Jeg vil ha en recap av de vanskeligste konseptene fra forelesning -> Repetisjonsgruppe!
- Oblig 2, frist Mandag 15.02 kl 23.59
- Meld dere på gruppe oblig 4!

De vanligste feilene på oblig 1

Array vs ArrayList

Array er raskere, men ArrayList har metoder, og man kan endre størrelse.

Så bruke Array hvis du vet antall elementer.

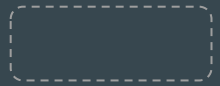
F.eks. i oblig1 så visste vi maks antall noder i nodelistene i array, og de fleste av dere (alle?) implementerte det slik at vi fylte opp et rack helt, før vi la til et nytt rack. Så de fleste racksene ville ha maskAntallNoder, bare det siste racket ville ha litt færre racks.

Datastrukturtegning

Anbefaler å lese hele dette dokumentet i sin helhet:

https://www.uio.no/studier/emner/matnat/ifi/IN1010/v21/notater/omdatastrukt-2021.pdf

Objekter skal ha heltrukket-linje-firkant: 

Alt som er statisk i en klasse skal inni en stiplet-linje-firkant: 

Husk å tegne objektene til array og arrayList som egne bokser!

While-loop vs for-loop vs for-each-loop

Vi bruker while hvis vi _ikke_ vet hvor mange ganger noe skal skje, ellers bruker vi for.
Altså: kan vi bruke for bruker vi for!

Skal vi gå gjennom elementer i en type liste (bokstaver i en string f.eks.) så bruker vi vanlig for hvis vi trenger indeksen til elementet, ellers bruker vi for-each.
Altså: kan vi bruke for-each på liste bruker vi for-each.

```
8  for (Katt katt : katter){¬
9  ··// Denne er mer leslig, men da har vi ikke tilgang til indeks¬
10 }¬
11 for (int indeks = 0; indeks < katter.size(); indeks++){¬
12 ··// Bruker denne hvis vi trenger indeksen¬
13 ··// Aksesserer elementet sånn for arraylist: katter.get(indeks)¬
14 }¬
15 for (int indeks = 0; indeks < katter.length; indeks++){¬
16 ··// Bruker denne hvis vi trenger indeksen¬
17 ··// Aksesserer elementet sånn for array: katter[indeks]¬
18 }¬
```

Du trenger ikke putte all koden din i en try/catch!

Det er bare new Scanner som kan gi error, så det er bare den vi trenger å putte inne i try.

Deklarerer variabelen fil utenfor så den er definert utenfor

```
Scanner fil = null;
try{
  fil = new Scanner(new File(filnavn));
} catch (Exception e) {
  System.out.println(e);
  System.exit(1);
}
```

Repetisjon forrige uke

Sist

Abstract class

.super

super()

Polymorfi

Ugg hjelp, skjønte ingenting av mentimeter-quizen i forelesningen!

Send meg direktemelding

1. Hva er super.?
2. Hva er super()?
3. Hva er polimorfi?
4. Hva printes på bildet?

Gir dere 3min på å svare

```java
class Dyr{
    public void lagLyd(){
        System.out.println("Hello");
    }
}
class Katt extends Dyr{
    public void lagLyd(){
        System.out.println("Mjau");
    }
}
class Hovedprogram{
    public static void main(String[] args) {
        Katt dyr1 = new Katt();
        Dyr dyr2 = new Katt();
        Dyr dyr3 = new Dyr();

        Dyr[] alleDyr = {dyr1, dyr2, dyr3};

        for (Dyr dyr : alleDyr){
            dyr.lagLyd();
        }
    }
}
```

Send meg direktemelding

1. Hva er .super?
    a. Kan brukes når vi vil aksessere noe i superklassen (f.eks. en metode)
2. Hva er super()?
3. Hva er polimorfi?
4. Hva printes på bildet?

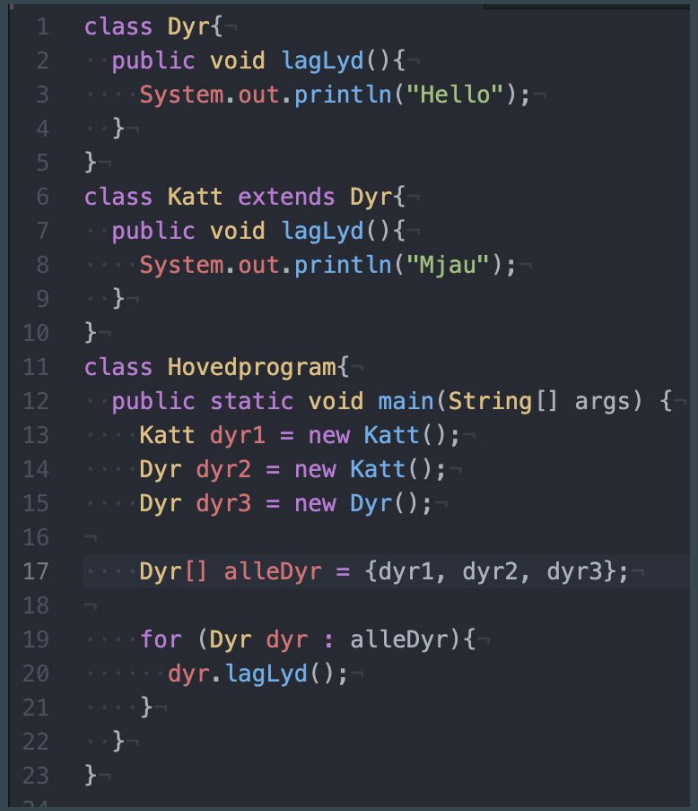
```java
class Dyr{
    public void lagLyd(){
        System.out.println("Hello");
    }
}
class Katt extends Dyr{
    public void lagLyd(){
        System.out.println("Mjau");
    }
}
class Hovedprogram{
    public static void main(String[] args) {
        Katt dyr1 = new Katt();
        Dyr dyr2 = new Katt();
        Dyr dyr3 = new Dyr();

        Dyr[] alleDyr = {dyr1, dyr2, dyr3};

        for (Dyr dyr : alleDyr){
            dyr.lagLyd();
        }
    }
}
```

Send meg direktemelding

1. Hva er .super?
   a. Kan brukes når vi vil aksessere noe i superklassen (f.eks. en metode)
2. Hva er super()?
   a. Kall på superklassen sin konstruktør
3. Hva er polimorfi?
4. Hva printes på bildet?



```java
class Dyr{
    public void lagLyd(){
        System.out.println("Hello");
    }
}
class Katt extends Dyr{
    public void lagLyd(){
        System.out.println("Mjau");
    }
}
class Hovedprogram{
    public static void main(String[] args) {
        Katt dyr1 = new Katt();
        Dyr dyr2 = new Katt();
        Dyr dyr3 = new Dyr();

        Dyr[] alleDyr = {dyr1, dyr2, dyr3};

        for (Dyr dyr : alleDyr){
            dyr.lagLyd();
        }
    }
}
```

Send meg direktemelding

1. Hva er .super?
   a. Kan brukes når vi vil aksessere noe i superklassen (f.eks. en metode)
2. Hva er super()?
   a. Kall på superklassen sin konstruktør
3. Hva er polimorfi?
   a. Når ulike klasser arver hverandre(extends) og de har metoder med samme signatur som gjør ulike ting. Så hvis vi kaller samme metode på ulike objekter kan de utføres ulikt.
4. Hva printes på bildet?

```java
class Dyr{
    public void lagLyd(){
        System.out.println("Hello");
    }
}
class Katt extends Dyr{
    public void lagLyd(){
        System.out.println("Mjau");
    }
}
class Hovedprogram{
    public static void main(String[] args) {
        Katt dyr1 = new Katt();
        Dyr dyr2 = new Katt();
        Dyr dyr3 = new Dyr();

        Dyr[] alleDyr = {dyr1, dyr2, dyr3};

        for (Dyr dyr : alleDyr){
            dyr.lagLyd();
        }
    }
}
```

Send meg direktemelding

1. Hva er .super?
   a. Kan brukes når vi vil aksessere noe i superklassen (f.eks. en metode)
2. Hva er super()?
   a. Kall på superklassen sin konstruktør
3. Hva er polimorfi?
   a. Når ulike klasser arver hverandre(extends) og de har metoder med samme signatur som gjør ulike ting. Så hvis vi kaller samme metode på ulike objekter kan de utføres ulikt.
4. Hva printes på bildet?

```
jonbon@jons-macbook-pro uke5 % java Hovedprogram
Mjau
Mjau
Hello
```

```java
class Dyr{
    public void lagLyd(){
        System.out.println("Hello");
    }
}
class Katt extends Dyr{
    public void lagLyd(){
        System.out.println("Mjau");
    }
}
class Hovedprogram{
    public static void main(String[] args) {
        Katt dyr1 = new Katt();
        Dyr dyr2 = new Katt();
        Dyr dyr3 = new Dyr();

        Dyr[] alleDyr = {dyr1, dyr2, dyr3};

        for (Dyr dyr : alleDyr){
            dyr.lagLyd();
        }
    }
}
```

Send meg direktemelding

1. Hva er .super?
   a. Kan brukes når vi vil aksessere noe i superklassen (f.eks. en metode)
2. Hva er super()?
   a. Kall på superklassen sin konstruktør
3. Hva er polimorfi?
   a. Når ulike klasser arver hverandre(extends) og de har metoder med samme signatur som gjør ulike ting. Så hvis vi kaller samme metode på ulike objekter kan de utføres ulikt.
4. Hva printes på bildet?

```
jonbon@jons-macbook-pro uke5 % java Hovedprogram
Mjau
Mjau
Hello
```

```java
class Dyr{
    public void lagLyd(){
        System.out.println("Hello");
    }
}
class Katt extends Dyr{
    public void lagLyd(){
        System.out.println("Mjau");
    }
}
class Hovedprogram{
    public static void main(String[] args) {
        Katt dyr1 = new Katt();
        Dyr dyr2 = new Katt();
        Dyr dyr3 = new Dyr();

        Dyr[] alleDyr = {dyr1, dyr2, dyr3};

        for (Dyr dyr : alleDyr){
            dyr.lagLyd();
        }
    }
}
```

Tre regler fra forelesning

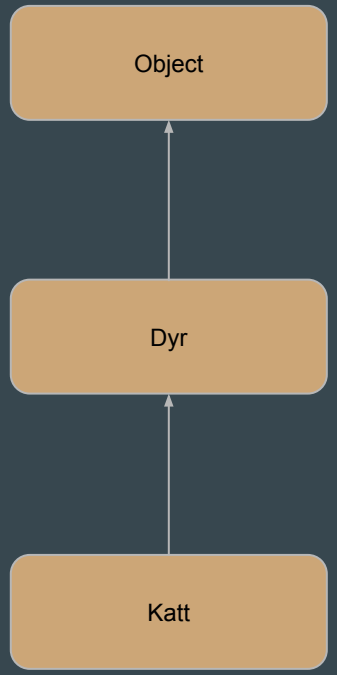
- Et kall på super **må** legges **helt i begynnelsen av** konstruktøren.
- Kaller man ikke super eksplisitt, vil Java **selv legge inn kall på super( )** helt først i konstruktøren når programmet kompileres.
- Hvis en klasse ikke har noen konstruktør, legger Java inn en tom konstruktør med kallet super();

Object, og metodene toString og equals

Object er eller klassers superklasse!

Og den har metoder man kan overskrive, bl.a. toString og equals!

Vi skriver om skrivUtInfo() i klassene våre.



Repetisjon denne uken

I dag

Interface

HashMap

Interface

Beskriver en gruppe objekter med de samme egenskapene.

Gjør bl.a. t vi kan putte alle de objektene (og bare de) i samme liste.

Et interface har definert signaturen til noen metoder. Alle klasser som implementerer et interface må ha disse metodene.

```
22  class Klassenavn extends Superklassenavn implements Interfacenavn{
23  ··// Her må vi ha med alle metoder som interfacet krever
24  }
```

```
22  class Klassenavn implements Interfacenavn{
23  ··// Her må vi ha med alle metoder som interfacet krever
24  }
```

Dyreregister

```java
abstract class Dyr{
    protected int alder;
    protected String navn;

    public Dyr(String navn, int alder){
        this.navn = navn;
        this.alder = alder;
    }
    public  int hentAlder(){
        return alder;
    }
    public  String hentNavn(){
        return navn;
    }
    @Override
    public String toString(){
        return "Navn: " + navn + "\n" + "Alder: " + alder;
    }
}
```

```java
class Hund extends Dyr{

    public Hund(String navn, int alder){
        super(navn, alder);
    }
    @Override
    public String toString(){
        return "Jeg er en hund!" + super.toString();
    }
}
```

```java
class Katt extends Dyr{

    public Katt(String navn, int alder){
        super(navn, alder);
    }
    @Override
    public String toString(){
        return "Jeg er en katt!" + super.toString();
    }

}
```

Dyreregister

Vi har en klasse Dyr. I den virkelige verden finnes det en del dyr som har sånn chip så man kan sjekke hvem som eier dem og sånn.

To eksempler er katter og hunder(tror jeg??)

Dyreregister

Vi har en klasse Dyr. I den virkelige verden finnes det en del dyr som har sånn chip så man kan sjekke hvem som eier dem og sånn.

To eksempler er katter og hunder(tror jeg??)

Vi skal lage et dyreregister der man kan slå opp på id og få info om dyr og eier.

```
jonbon@jons-macbook-pro uke5 % java Dyreregister
Velkommen! Hva er id til dyret du har funnet?
123
123
Beklager vi fant ikke dyret du lette etter
Velkommen! Hva er id til dyret du har funnet?
32424543534
Vi fant dyret du leter etter!
Navn: Max
Alder: 15
Eier:Navn: Maria alder: 15
```

HashMap

Hvis vi har en mengde objekter kan vi bare putte dem i en liste og gå gjennom listen for å finne det vi vil ha.

```
String id = "1234"
HarChip dyretViLeterEtter = null;
for (HarChip dyr : dyreregister){
  if dyr.hentId() == id{
    dyretViLeterEtter = dyr;
  }
}
```

Men det går veldig mye raskere, og ser veldig mye penere ut å bruke en hashmap(ordbok) (hvis vi ofte slår opp på samme verdi):

```
String id = "1234"
HarChip dyretViLeterEtter = dyreregister.get(id);
```

Interface

Uansett om vi bruker Array, ArrayList eller HashMap må vi fortelle programmet hva vi vil putte oppi.

```
34
35      NavnKlasseEllerInterface[] navnVariabel = new NavnKlasseEllerInterface[lengdeArray];
36      ArrayList<NavnKlasseEllerInterface> navnVariabel = new ArrayList();
37      HashMap<NavnKlasseEllerInterfaceNokkel, NavnKlasseEllerInterfaceVerdi> navnVariabel = new HashMap();
38
```

Interface

Uansett om vi bruker Array, ArrayList eller HashMap må vi fortelle programmet hva vi vil putte oppi.

```
34
35      NavnKlasseEllerInterface[] navnVariabel = new NavnKlasseEllerInterface[lengdeArray];
36      ArrayList<NavnKlasseEllerInterface> navnVariabel = new ArrayList();
37      HashMap<NavnKlasseEllerInterfaceNokkel, NavnKlasseEllerInterfaceVerdi> navnVariabel = new HashMap();
38
```

Så det vi vil:

```
//key: id, value: dyr som harChip: Hund og Katt
HashMap<String, Hund&Katt> dyreregister = new HashMap();
```

Interface

Uansett om vi bruker Array, ArrayList eller HashMap må vi fortelle programmet hva vi vil putte oppi.

```
34
35    NavnKlasseEllerInterface[] navnVariabel = new NavnKlasseEllerInterface[lengdeArray];
36    ArrayList<NavnKlasseEllerInterface> navnVariabel = new ArrayList();
37    HashMap<NavnKlasseEllerInterfaceNokkel, NavnKlasseEllerInterfaceVerdi> navnVariabel = new HashMap();
38
```

Så det vi vil:

```
//key: id, value: dyr som harChip: Hund og Katt
HashMap<String, Hund&Katt> dyreregister = new HashMap();
```

For å gjøre dette lager vi et interface:

```
//key: id, value: dyr som harChip: Hund og Katt
HashMap<String, HarChip> dyreregister = new HashMap();
```

Dyreregister

Vi har en klasse Dyr. I den virkelige verden finnes det en del dyr som har sånn chip så man kan sjekke hvem som eier dem og sånn.

To eksempler er katter og hunder(tror jeg??)

Vi skal fikse sånn at Katt og Hund blir Dyr med chip!

```
1  interface HarChip{
2      Person hentEier();
3      String hentId();
4  }
```

```
class Katt extends Dyr implements HarChip{
```

```
class Hund extends Dyr implements HarChip{
```