IN1010 - Seminar 5

- Interface

Laget av: Marlen Jarholt (marleja@ifi.uio.no) og Johanna Haarseth (johannph@ifi.uio.no)

Praktisk

Meld dere på gruppe oblig 4, se emnesiden!
Hvis dere allerede vet hvem dere vil samarbeide med holder det å lage gruppe i devilry.

Husk å sjekke emnesiden regelmessig

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!

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!

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

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.

```
for (Katt katt : katter){
    // Denne er mer leslig, men da har vi ikke tilgang til indeks
}
for (int indeks = 0; indeks < katter.size(); indeks++){
    // Bruker denne hvis vi trenger indeksen
    // Aksesserer elementet sånn for arraylist: katter.get(indeks)
}
for (int indeks = 0; indeks < katter.length; indeks++){
    // Bruker denne hvis vi trenger indeksen
    // Aksesserer elementet sånn for array: katter[indeks]
}
```

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

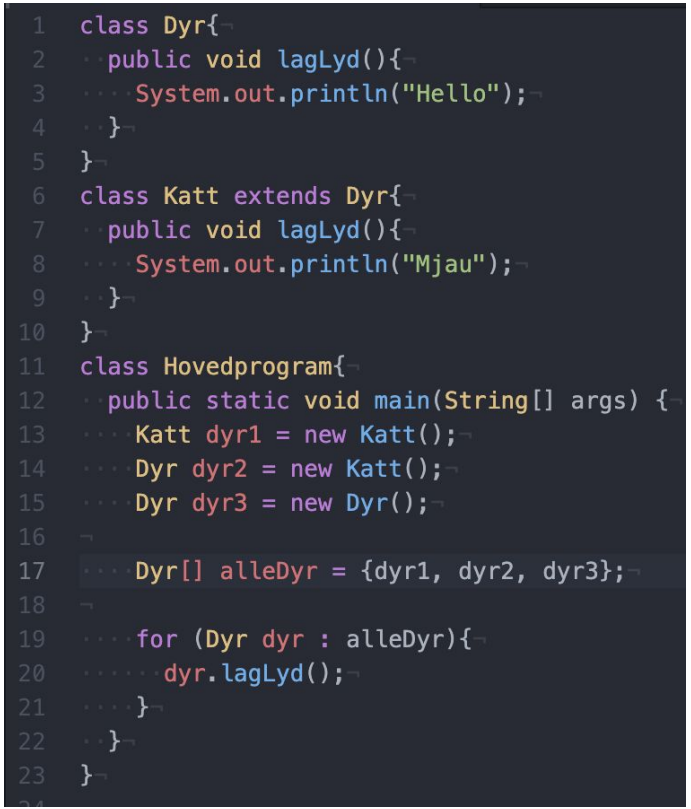
```
Scanner fil = null;
try{
    fil = new Scanner(new File(filnavn));
} catch (Exception e) {
    System.out.println(e);
    System.exit(1);
}
```

Repetisjon forrige uke

Repetisjon

Send meg en direktemelding!

Hva printes her?



```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();
    }
  }
}
```

Repetisjon

Send meg en direktemelding!

Hva printes her?

```
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();
    }
  }
}
```

Interface

Beskriver en gruppe objekter med de samme egenskapene.

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

Gjør bl.a. at vi kan putte alle de objektene (og bare de) i samme liste/HashMap etc.

```
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  }
```

Repetisjon denne uken

Send Marlen en direktemelding i chatten

Vil du jobbe sammen med noen andre ? (ja /nei)

Hvis du har noen ønsker på hvem du vil jobbe med, så send det i samme melding

Svar gjerne også om svaret skulle være nei

Beholder

Et objekt som representerer en gruppe objekter av samme type.

Ofte metoder for å legge til, hente ut og finne størrelse.

Eksempler på beholdere: array(har ingen metoder), ArrayList, HashMap, Lenkeliste

Implementere lenkeliste

Vi trenger:

Beholderne i lenkelisten, f.eks. noder
- kunne ta vare på data
- kunne ha en referanse til neste beholder/node

Generisk klasse

Gjøre om Node til en generisk klasse.

Se løsningsforslag ukesoppgavene for kode.

Generisk klasse

(Jeg brukte T i register, men dette er standarden:)

E - Element (used extensively by the Java Collections Framework)

K - Key

N - Number

T - Type

V - Value

Ris, ros, forslag ?

https://nettskjema.no/a/180345

Breakoutrooms

1. Slå på kamera og ha en presentasjonsrunde
2. Diskuter:
    a. Hva er forskjellen på interface og superklasse?
    b. Hva er fordelene på hashmap(ordbok)? Når burde man bruke det?
3. Jobb sammen med ukesoppgavene, de ligger på emnesiden -> grupper
    a. Enten ved at én deler skjerm med codecollab.io/
    b. OBS: codecollab er gratistjeneseter som UiO ikke har avtale med, sannsynligvis vil de samle data om dere. Dere kan fint løse oppgavene uten å bruke disse tjenestene!
4. Bruk “ask for help”-knappen for å få hjelp 😊
5. Vi møtes her igjen for å gå gjennom oppgavene til slutt (dere bestemmer hvilke)

Jobbe med oppgaver

ArrayList VS Array (Oppgave 1)

ArrayList

- Dynamisk
- Kan ikke ha primitvietype/kun objekter

Likheter

- Si hvilken type/klasse du skal putte inni
- Beholdere begge to

Array

- Statisk
- Klasser/objekter og primitvietyper

ArrayList VS Array (Oppgave 1)

ArrayList

- Bruker mer tid
- Bruker mer minne
- Kan ikke ta vare på primitive typer
- Har ikke en fast størrelse, dynamisk
- Ikke lagret etter hverandre i minne

Likheter

- Kan kun ta vare på objekter av samme type
- Er begge beholdere som strukturer objekter
- Du kan bruke for-looper over begge

Array

- Mindre tid
- Mindre minne
- Kan ta vare på primitive typer
- Har en fast størrelse (statisk)
- Lagret etter hverandre i minne