IN1010 Gruppetime

Julian Fjeld
og
Sivert Fjeldstad Madsen

Introduksjon

- Julian Fjeld – julianfj@uio.no
- 2. år Programmering og Systemarkitektur
- Sivert
- Inspirasjon – Wave Function Collapse
  https://youtu.be/20KHNA9jTsE?t=395
  https://github.com/mxgmn/WaveFunctionCollapse

Våre gruppetimer

- Også kalt seminartimer, undervisningstilbud
- Gå gjennom pensum for uka
- Ha åpne diskusjoner om pensum og faget
- Stille spørsmål, få greie på ting som ikke ble tydelig i forelesning
- Erfaring
- Parprogrammere ukeoppgavene
- Livekoding og gjennomgang av ukeoppgavene
- Ikke stedet for å jobbe med obliger

Nyttig info

- Emnesiden IN1010
- https://www.uio.no/studier/emner/matnat/ifi/IN1010/

Obliger

- Innleveringer skjer i devilry - https://devilry.ifi.uio.no/
- Trenger du utsettelse sender du ***mail*** til din retter, ikke beskjed i devilry
- Om du ikke får obligen godkjent kan du ha fått et nytt forsøk med first om 3 virkedager, følg med
- Juks og plagiat slås hardt ned på – utestengning fra Universitetet i ett år
- Selv om man har fått godkjent obliger tidligere år må man fortsatt få alle 7 obliger godkjent
- Snakk gjerne om forskjellige løsninger med hverandre, ikke del kode

ForVei

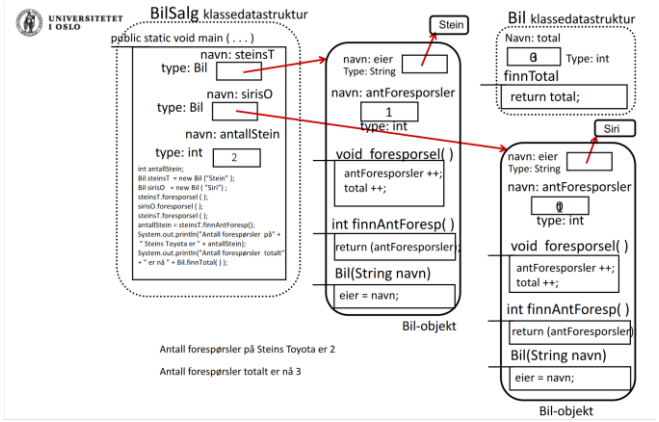
- Om du trenger noen å snakke med om livet som student
- https://www.mn.uio.no/studier/forvei/

Si fra!

- Om du trenger å varsle eller vil gi Universitetet tilbakemelding
- Gode ting
- Dårlige ting
- Farlige ting
- Fæle ting
- https://www.uio.no/studier/kontakt/si-fra/

IN1010

- Praktisk programmeringsfag
- Begynner enkelt, noen ting kan virke meningsløst eller som bortkastet tid



- Stick with it!
- Det kommer på eksamen
- Mye jobb, mange obliger
- Om man føler at man havner bakpå, bruk ressursene som er tilgjengelige

Java

- Alt i java må være i klasser
- Programmer kan bli store og inneholde mange filer med klasser. Alle disse kan kompileres samtidig med kommandoen:
  javac *.java
- Hovedprogrammet må ligge i en klasse og er en statisk metode som må se ut på en helt spesiell måte:

```
public static void main(String[] args){
        // Innhold
        // Opprett objekter av andre klasser og
        // gjør litt av hvert
}
```

- Når man skal kjøre programmet bruker man kommandoen java og navnet på klassen som inneholder main-metoden. F.eks:
  java Hovedprogram
- Denne kommandoen gjør at man kjører main-metoden i klassen Hovedprogram som er definert i filen Hovedprogram.java i dette spesifikke eksempelet

Java

- Static

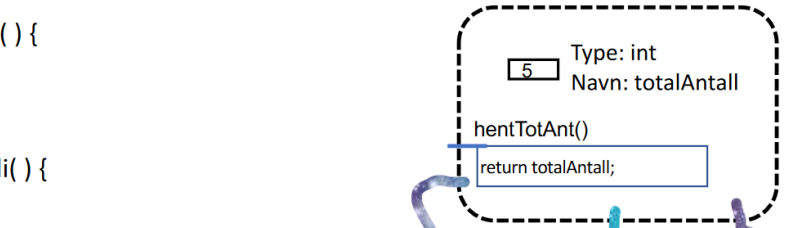
UNIVERSITETET I OSLO

```
class Teller {
    private static int totalAntall =0;
    private int verdi;
    public static int totalAntall ( ) {
        return totalAntall;
    }
    public void tellOpp( ) {
        verdi ++;
        totalAntall++;
    }
    public int hentVerdi( ) {
        return verdi;
    }
}
```

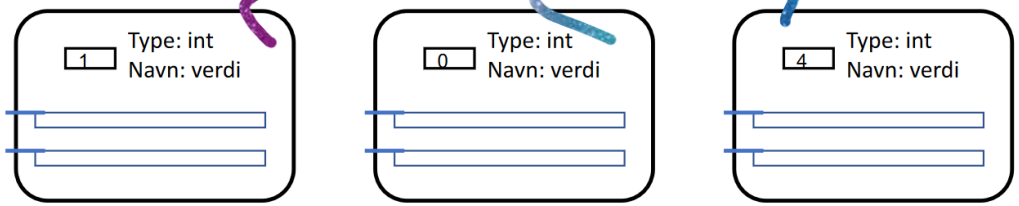
static-egenskaper

Institutt for informatikk

Når programmet starter lages en *klasse-datastruktur* av alle "static"-egenskapene til alle klassene i programmet



Etter f.eks. 3 kall på new Teller() har vi 3 objekter , og etter 5 kall på tellOpp() har instansvariablene f.eks. disse verdiene:



De tegningene vi har sett hittil av variable, metoder, objekter og klassedatastrukturer kaller vi til sammen Java **datastrukturer**

Java

- Pekere/ referanser
- Metodeinstans
- Tegninger