Gruppetime uke 11

IN1000, gruppe 7 - høst 19

Dagens temaer

- Delegering av ansvar i klasser
- Nyttige ting for Oblig 8
- Fortsette på studentsystemet fra forrige uke

Delegering av ansvar i klasser

Fra forelesningen tirsdag 29.10:

- Høy kohesjon
  - Klasser skal ha ansvar for sitt, og bare sitt område av programmet
  - Aggregering av kode
  - Vi streber alltid etter **høy kohesjon**
- Lav kobling
  - Innviklede system er vanskelige å navigere
  - Analyser hvilke klasser som behøver å kommunisere med hverandre
  - Vi streber alltid etter **lav kobling**
- Lav kobling + høy kohesjon = ❤️

Delegering av ansvar i klasser

Eksempel:

- Klassene Sang og Spilleliste fra Oblig 7:
  - **Kohesjon**: Hvilke arbeidsoppgaver er det naturlig at de forskjellige klassene har?
    - **Sang**: holde på informasjon om tittel og artist, metoder for å spille av sang, osv.
    - **Spilleliste**: holde på informasjon om navnet på listen, holde på listen over Sang-objekter, metoder som skal gjennomføres på alle Sang-objekter samtidig
    - Bygger programmene nedenfra, bruker de ovenfra
  - **Kobling:** Hvilke klasser må kommunisere med hverandre?
    - Spilleliste-objektet må holde på referanser til Sang-objekter
    - Sang-objekter trenger ikke nødvendigvis vite hvilken spilleliste de tilhører

Nyttige ting for Oblig 8

Hvordan starte på en stor oblig med laaang oppgavetekst?

- **Før** man begynner å programmere:
  - Les gjennom **hele** obligteksten
  - Hvilke klasser skal være med? Skriv inn kodeskjelleter
  - Les hver deloppgave **hver for seg**
    - Hva er poenget med denne metoden? Hvordan skal den endre objektet? Hva skal den returnere? Osv.
    - Skriv inn **pseudokode** basert på oppgaveteksten, eksempel: _generer
- Bygg opp programmet nedenfra og opp
  - Programmer de minste klassene først, deretter klassene som skal bruke de mindre klassene, og så videre.
  - Test én og én metode i klassen før du går videre til neste klasse

Nyttige ting for Oblig 8

- Datastrukturtegning av spillebrettet
- Kennel-programmet :
  - Under uke 9, her.
  - Relevant for generer, finnNabo, tegnBrett
  - Idag: tegne opp kartet med hunder

Nyttig for oblig 8

Hack for å klarere terminalen

```
#Tips: importer os, så klarerer vi terminalen
import os

#For å klarere skjermen kan vi lage denne
def cls():
    os.system('cls' if os.name=='nt' else 'clear')

#For hver gang vi vil klarere terminalen
#kan vi kalle den slik:
cls()
```

Gruppearbeid

Skrive pseudokode for *oppdatering()*

Gå sammen i grupper på 2-3 personer, og skriv pseudokode for metoden oppdatering() i klassen Spillebrett, basert på oppgaveteksten

Ca 10 min.

5. For å beregne neste generasjon av celler trengs metoden *oppdatering*. Denne metoden skal gjøre følgende:
    a. Opprett to lister. Den ene listen skal inneholde alle døde celler som skal få status “levende”, mens den andre skal inneholde levende celler som skal få status “død”. La listene være tomme foreløpig.

    b. Deretter skal metoden gå gjennom rutenettet ved hjelp av en nøstet løkke. For hver celle skal den sjekke om cellen er levende eller død og deretter beregne om den skal endre status på bakgrunn av antallet levende naboer. Her blir du nødt til å hente opp alle naboene til en celle og telle antallet som lever. Følg reglene beskrevet under “Spillets regler” lenger opp. Celler som skal endre status skal legges inn i den riktige av de to listene vi laget tidligere.

    c. Først når alle cellene er sjekket og listene er fylt med Celle-objekter skjer selve oppdateringen, og objekter i de to listene endrer status ved hjelp av Celle-metodene *settLevende* eller *settDoed*.

    d. Til sist må du huske å oppdatere telleren for antall generasjoner.

Fortsette på studentsystemet

- fag.py → forrige uke
- student.py → forrige uke
- studentsystem.py → idag
- hovedprogram.py → idag

Student

Variabler:

- Navn
- Fagliste

Metoder:

- leggTilFag(fag)
- hentAntallFag()
- hentStudentNavn()
- skrivFagPaatudent()

Fag

Variabler:

- Navn
- Studentliste

Metoder:

- LeggTilStudent(student)
- hentAntallStudenter()
- hentFagNavn()
- skrivStudenterVedFag()

Lykke til med oblig 8 - sees neste uke :)