IN1000 – GRUPPE 11

Uke 10

AGENDA

- Status på innleveringer
- Eventuell repetisjon
- Sammenhengende oppgaver
  - Jobber videre neste uke

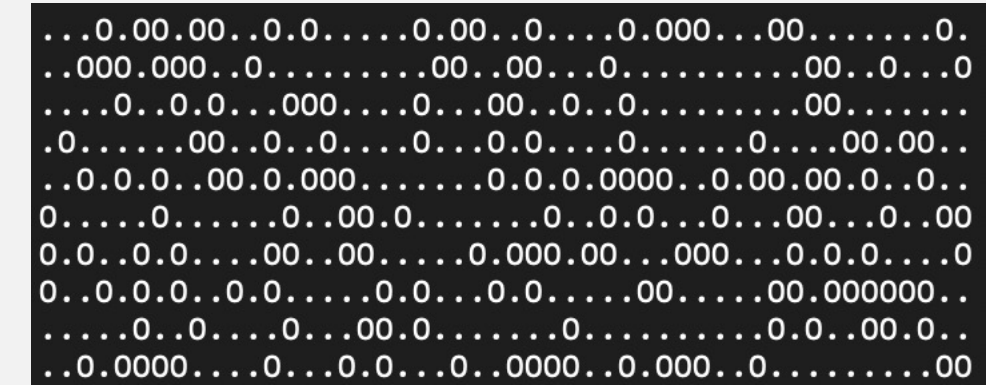
OBLIG 7

- Tanker om hvordan den har vært?
- Utvidet frist: pass på å få levert i løpet av torsdagen
  - Gjerne før
- Denne må man bestå!
  - Viktig å følge med på epost slik at man vet om man har bestått eller må levere andre forsøk
    - Husk at selve tilbakemeldingen bare finnes på Devilry-siden

INNLEVERING 8

- Frivillig
  - Anbefales selvfølgelig!
- *(Conway's) Game of Life*
  - En simulering der cellers overlevelse baseres på cellene rundt seg selv
    - Celler kan dø og gjenopplives
  - Vi bruker nøstede lister
    - Husk at en liste fint kan plasseres inni en annen liste
- Video: https://www.youtube.com/watch?v=xP5-ileKXE8
  - I IN1000 bruker vi bare terminalen

Slik kan det se ut.
O = levende celle
. = død celle



MENTI OM HVA VI SKAL FOKUSERE PÅ

OPPGAVE

Studenter og Fag

Denne uken skal dere begynne å jobbe på et større program. Dere vil denne uken lage klassene “Fag” og “Student” som begge skal ha referanser til hverandre. Neste uke vil dere jobbe med å knytte dem sammen til et større system.

(Oppgaven ligger tilgjengelig som PDF og Word-dokument under gruppe 11, uke 10 på emnesiden)

OPPGAVE 1

1.1 Lag en klasse “Fag”, som skal ha et navn og en liste som holder på alle studentene som tar det aktuelle faget. Ved opprettelsen av et fag vil denne studentlisten være tom.

1.2 Lag en metode i Fag-klassen som heter “legg_til_student(student)” som tar imot en Student som parameter og legger den til i listen. Hvor i listen du legger den til spiller ingen rolle, så du må gjerne bruke .append().

1.3 Lag en metode i Fag-klassen som **returnerer** antallet studenter som tar faget, kall denne “hent_antall_studenter()”.

1.4 Lag en metode som **returnerer** fagets navn, hent_fag_navn().

1.5 Lag en metode “skriv_studenter_ved_fag()” som skriver ut fagnavnet, og deretter alle studentene som tar faget.

Merk: Anta at metoden hent_student_navn() er implementert Student-objektet.

OPPGAVE 2

2.1 Lag en klasse “Student”. Studenten skal ha et navn og en liste som holder på fagene studentene tar. Ved opprettelsen av en student vil denne faglisten være tom.

2.2 Legg til en metode i Student-klassen som heter “legg_til_fag(fag)” som tar imot et fag som parameter og legger det til i studentens liste over fag.

2.3 Lag en metode i Student-klassen som **returnerer** antallet fag studenten tar, kall denne “hent_antall_fag()”.

2.4 Lag en metode i Student-klassen som **returnerer** studentens navn, hent_student_navn().

2.5 Lag en metode “skriv_fag_paa_student()” som skriver ut studentens navn, og deretter alle fagene som studenten tar.

Her må du bruke hent_fag_navn() fra Fag-objektet.

OPPGAVE 3

3.1 Lag et hovedprogram som tester litt funksjonalitet.

3.2 Tegn opp datastrukturen til første del av hovedprogrammet.

OPPGAVE 4

Lag et lite testprogram hvor du lager noen studenter og noen fag. Legg til fagene i studentenes liste og studentene inn i fagene. Skriv ut hvor mange fag studentene dine tar, og hvor mange studenter hver av fagene dine tar.