

Deltakelse på undervisningsaktiviteter ved Ifi

Oppgaven som PDF finner du også på semestersiden til IN1000

Et vanlig problem for mange emner, er at oppmøte til grupper og aktiviteter endrer seg i løpet av semesteret. Noen grupper får flere studenter enn planlagt, mens andre grupper får færre. I denne oppgaven skal vi lage et system som hjelper undervisere med å undersøke oppmøtet i de ulike gruppene og aktivitetene for emnene vi har på Ifi. På den måten kan undervisningen planlegges bedre.

Gjennom hele oppgaven står du fritt til å innføre egne metoder, variabler og eventuelt klasser, der du selv ser behovet for dette slik du løser oppgaven.

Oppgave A (2 poeng)

Skriv klassen `Student`. Ved opprettelse av objekter av klassen, skal det opprettes en tom liste over emner studenten følger. Konstruktøren skal ta inn studentens brukernavn, og lagre dette i en instansvariabel.

Du skal skrive én metode i klassen, `hentBrukernavn(self)` som skal returnere studentens brukernavn.

Oppgave B (4 poeng)

Du skal også skrive klassen `Emne`. Ved opprettelse av objekter av klassen, skal en tom liste over aktiviteter (grupper o.a.) tilknyttet emnet opprettes. Konstruktøren skal ta inn kode på emnet, og lagre dette i en instansvariabel.

Klassen skal ha én metode. Denne metoden skal legge en aktivitet til listen over aktiviteter.

Oppgave C (12 poeng)

Du skal opprette en klasse `Dato`. Denne klassen skal ta tre heltall som argumenter i sin konstruktør: `dag`, `maaned` og `aar`. År skal representeres ved to siffer; 2020 vil for eksempel representeres som 20. Du kan anta at programmet kun skal brukes etter år 2000 og før år 2100. Klassen skal ha følgende metoder:

- `absoluttDato(self)`, som returnerer datoen som ett heltall, der rekkefølgen blir år, måned og dag. Dette sikrer at man alltid kan hente ut datoer kronologisk. For eksempel vil 19. november 2020 representeres som 201119. Merk at du kan trenge å legge til ekstra nuller for å få tallet på seks siffer.

- `__str__(self)`, som returnerer datoen på et pent og leselig format, som en streng. 19/11-20 vil kunne returneres som "19. november 2020". Du kan anta at programmet kun brukes i høstsemesteret, altså trengs bare september-desember.

Oppgave D (12 poeng)

Klassen Aktivitet representerer en aktivitet knyttet til et emne. Dette er typisk en gruppetime eller lab-øvelse.

Du skal skrive klassen Aktivitet. Ved opprettelse skal konstruktøren ta følgende parametre:

- *emne*: et objekt av klassen Emne, som sier hvilket emne aktiviteten er tilknyttet.
- *dato*: et objekt av klassen Dato, riktig dato for aktiviteten.
- *aktivitetsnummer*: et heltall som indikerer hvilket nummer aktiviteten har.

Konstruktøren skal også opprette to tomme lister, den ene for studenter som er registrert på aktiviteten i StudentWeb, og den andre for studenter som faktisk har møtt opp til denne aktiviteten (oppmøteregistrering).

Klassen skal inneholde følgende metoder:

- *leggTilRegistrertStudent(self, student)*: legger et student-objekt til listen over registrerte studenter.
- *registrerOppmote(self, student)*: studenten har møtt opp til aktiviteten, og objektet legges til listen for oppmøtte studenter.
- *skrivUtOppmotteStudenter(self)*: skriver ut navnet på alle studenter som har møtt til aktiviteten.
- *absoluttDato(self)*: returnerer dato for aktivitet på absolutt-format, som beskrevet i Dato-klassen.
- *oppmote(self)*: returnerer antall oppmøtte studenter til aktiviteten.
- *__str__(self)*: returnerer en streng som inneholder aktivitetens emne, nummer og antall oppmøtte til aktiviteten.

Oppgave E (3 poeng)

Lag en klasse Undervisningsadministrasjon. Klassen skal ikke ta noen argumenter til sin konstruktør. Ved opprettelse av objekter, skal følgende instansvariabler opprettes:

- *emner*: en ordbok, der en streng med emnekode vil være nøkkel, og objekter av Emne skal være verdi.
- *datoer*: en ordbok, dato på absolutt format (som er lik returverdien til *absoluttDato*) skal være nøkkel, mens Dato-objekter er innholdsverdi.
- *studenter*: en ordbok over studenter. Brukernavn er nøkkel, Student-objekt er verdi.

Oppgave F (10 poeng)

Lag en metode i Undervisningsadministrasjon som leser inn filer og oppretter emner og deres aktiviteter. Merk at det kun skal være ett dato-objekt for hver dato. Det skal også kun være ett objekt per emne. Filen som skal leses vil være på følgende format (eksempel):

IN1000 Gruppe1 20 09 18
IN1000 Gruppe2 20 09 22
IN1000 Gruppe3 20 10 08
IN5090 Lab1 21 01 12

Obs: husk at hvert emne kun skal representeres som ett objekt, og at aktiviteter legges til dette objektet. Objektene legges til sine respektive ordbøker.

Oppgave G (7 poeng)

Lag en metode i Undervisningsadministrasjon som leser informasjon fra fil om studenter, deres emner og hvilke aktiviteter de er registrert for. Dataene du leser, skal legges inn i datastrukturen, og Student-objekter skal opprettes for hver enkelt student. Et eksempel kan være som følger:

henrihlo IN1000 Gruppe1
henrihlo IN1010 Gruppe4
siriamj IN2010 Gruppe3
siriamj IN2010 Lab1

Du kan anta at emnet eksisterer i ordboken over emner, og at gruppe/aktivitet eksisterer i emnenes lister over aktiviteter.

Oppgave H (5 poeng)

Skriv en metode i Undervisningsadministrasjon *skrivGrupperMedLavtOppmoete(self, antall)* som skriver ut en liste over de gruppene som har hatt færre enn et antall oppmøtte studenter gitt som parameter.