



Dette er klassediagrammet til oppgave 4 fra eksamen høst 2019 som vi ender opp med utifra det som står eksplisitt i oppgaveteksten (etter litt hopping fram og tilbake) eller det vi kan anta utifra den (f.eks. at klassen "Emne" skal ha instansvariabelen "obliger"). Se etter og bruk mønstre som går igjen for å enkelt overføre data fra en metode til en annen, og tenk på hvordan klassene burde lagre objekter av hverandre for å kunne bruke hverandres metoder.

Legge merke til at dette klassediagrammet ikke følger alle reglene for UML til punkt og prikke. Et av mange eksempler på dette er at man ikke skal spesifere navn på nøkkelverdier og innholdsverdier i ordbøker, men jeg har gjort dette for å tydeliggjøre hva dataene representerer.

Når dere har eksamen kan dere tegne et klassediagram med blyant og papir hvor dere sørger for at det er god plass til å gjøre forandringer på den senere.

Eller så kan dere lage et klassediagram i form av en hierarkisk liste i et skriveprogram som Word eller lignende. Den kan se sånn ut:

- Emne
  - - \_emnekode: str
  - - \_studenter: dict[?: ?, student: Student]
  - - \_rettere: list[retter: Retter]
  - - \_obliger: ?
  - -----
  - + administrer(): null
  - - \_opprettOblig(): null
  - - \_startRetting(): null
  - - \_skrivEksamensliste(): null

Uansett hvilken metode dere velger, ikke tenk på om du bryter reglene for klassediagram og bare bruk en notasjon som gir mening for deg, vær veldig nøye men ikke bruk for mye tid på den heller. Bare pass på at dere spesifiserer navn, type, og private/public for alle verdier og navn, parameterverdier, returverdier, og private/public for alle metoder.

Husk at hovedpoenget er å få nok oversikt og innsikt til å se hvilke metoder som kan brukes av andre metoder, og hvilke typer og navn verdiene dine burde ha for å holde programmet konsekvent. Tegn kun klassediagram for den største oppgaven hvis dere føler at dere trenger det og føler at dere har tid til det.