Prog 2 Oblig 6 raport

# ULM

Jeg har valgt å ta med CelestialBody, NaturalSatellite, Planet, Moon, Star og PlanetSystem. Dette er siden jeg har mesteparten av koden fra før.

Jeg lagde også klassen FileRW (File Read Write), mest for å gjøre lesing og skriving lettere. Jeg la også til en fungsjon for å se om en fil eksisterer slik at jeg ikke får feil når jeg prøver å lese fra en fil som ikke finnes.

Jeg valgte å ikke ta med settere og gettere i diagrammet. Jeg tror ikke man trenger det siden det ikke står spesefikt noen steder at det må være med.

# Kort overordnet forklaring

For å løse oppgaven begynte jeg å tegne opp ULM diagrammet for å se hvilke klasser jeg trengte. Når jeg hadde funnet de klassene jeg trengte og noen ekstra klasser som kan gjøre bonus oppgavene litt bedre begynte jeg å lage de klassene.

Etter jeg hadde lagd alle klassene lagde jeg en main og en application, main brukte jeg til å generere 7 observasjoner. Når jeg hadde testet at jeg kunne generer alt jeg trengte for 7 observasjoner tokk jeg å implementerte skrive fungsjonene. Jeg lagde skrivefungsjon til CSV og JSON. Når jeg lagde disse fungsjonene fant jeg ut at jeg ville lage headeren mer «dynamisk» generert, dette for at hvis jeg endret noe i en klasse trengte jeg bare å endre på en fungsjon i den klassen. Jeg lagde også en fungsjon for å skrive ut informasjonen fra vær klasse. Dette er fungsjonene toCSV() og toCSVFormat().

Når jeg da viste at jeg kunne skrive til fil, lagde jeg en fungsjon for å lese en fil. Her brukte jeg noe jeg har gjørt i de forrige obligene, med å ha en setProp() fungsjon i alle klasser jeg har lyst til at skal bli lest fra. Jeg lagde da et interface (ICSVRead) som har setProp() som en fungsjon og satt Observation, Location, CelestialBody og Animal til å implementere interface. Mållet med denne fungsjonen er at for å registrerer en verdi på et objekt trenger jeg bare å sende in 2 strenger, en med hvilken verdi jeg vil endre og en med verdien. Så når jeg leser en CSV fil kan jeg gjøre dette enkelt med å gå en og en kolonne bort og lese navnet på verdien (navn, id, osv.) og bare sende in en verdi.

Når jeg kunne lese og skrive begynte jeg med å lage repository og controlleren. Jeg lagde fleste fungsjonene jeg tror jeg trenger i begge repositoryene (en til CSV og en til JSON), men jeg har bare lagd de jeg trenger i selve controlleren.

For å oppdatere objekter lagde jeg en update fungsjon. Denne tar inn en av samme objekt og legger endrer orginal objektet til de definerte verdiene i det «nye» objektet.

Jeg kopierte da vue filen fra oppgave siden og implementere den, og endrer litt på CSS’en. Det tokk litt tid for å få den til å fungere, siden jeg hadde lagt til noen ting som ikke fungerte, men når jeg fant ut av det fungerte siden. Jeg satt sefølgelig også opp application slik at jeg kunne starte javalin siden.

# Hjelp av

Jeg brukte mine gamle prosjekter, forelesning 14, 15, 16, pluss google (mye StackOverflow, men litt andre kilder også). Alle kildene står i tekst dokumentet Sources.txt, hvis jeg ikke har husket å legge de til i dette dokumentet også. Jeg har ikke vært på noen øvingstimer eller spurt noen om veiledning.