

**Forklar med dine egne ord de klassene du har identifisert, og relasjonen mellom dem.**

Observasjon er da hovedklassen. Observasjon inneholder 8 instansevariabler, der to av de kommer fra andre klasser. De to klassene er lokasjon og animal.

Lokasjon inneholder 5 instansevariabler der to av de også er fra andre klasser. Disse to klassene er Planet og Biom. De inneholder bare en instansevariabel. Har valgt at planet og biom er klasser, fordi det finnes flere lokasjoner per planet, som har forskjellige bioms.

Animal har bare 2 instanvariabler, men siden det bare var 3 typer Animal som var oppdaget med tanke på oppgaveteksten, så antar vi at Animal klassen ikke kommer til å ha noen egne objekter. Derfor setter vi den som abstract. Virvelløse, Amfibier og fugler vil dermed arve fra Animal. De arves pga de kommer til å både ha navn og latinnavn. Slipper dermed å skrive det på nytt i hver subklasse.

ObservationRepository er en klasse som vil holde på dataene vi får fra Observasjon klassen, som vil få en implementasjon fra IRepository interfacet. Dette vil tvinge klassen til å override metodene som er i IRepositoryen. ObservationJSON vil da hente dataen som ObservationRepositoryen holder på(Lagret i dens konstruktør) og både skrive og lese til json.

ObservationControlleren har så en relasjon til IRepository, som vil da bruke metodene der de har blitt implementert til å koble model klassene(observasjon, lokasjon(planet, biom),animal(virvelløs, amfibier, fugler)) med viewet.

* **En kort overordnet forklaring av hvordan du har løst oppgavene**
  + **Er det noe spesielle valg du har tatt?**
  + **Har du laget noen "hjelpemetoder" for å gjøre koden mer oversiktlig?**
  + **etc.**

Har løst oppgavene ved å tenke slik:

3.2 modeller 🡪

Valget av sorteringsmetode: Tenkte at man både kunne få nytte av å sortere Observasjon, lokasjon og Animal på navn. Det er fordi de inneholder forskjellige variabler, og man kan dermed velge hvordan man vil sortere de ved hjelp av anonyme klasser.

Valgte å lage Planet og Biom som egne klasser da det finnes mange forskjellige Biomer og planeter. Det er dermed lettere å gjenbruke disse ved å kalle dens objekt, pga det kan finnes flere observasjoner per planet og biom.

Lagde en observationRepository som holder på dataene om biom, planet, lokasjon og observasjon i sin konstruktør samt et filnavn. La den inn i observationJSON som en instansevariabel og brukte den i new file(this.obsRep.getFilnavn()) i skrivfil og readvalue i lesJsonFil.

3.4 Filskriving 🡪

Med tanke på CRUD, så lagde jeg en metode i observationRepository for å hente ut en observasjon med id som parameter. Det er fordi id ser vi på som en unik datatype.

* **Hvilke deler av applikasjonen har dere fått hjelp til eller funnet informasjon om på nettet eller i forelesningseksemplene (NB! det er ikke negativt å klare å benyttes kode fra andre kilder. Det er imidlertid veldig negativt å ikke opplyse om dette). Husk også å skrive opp de større tingene dere har fått hjelp til på øvingstimer/veiledning. Dette inkluderer også de faktiske referansene til kildene.**

Referanser:

<https://stackoverflow.com/questions/28802544/java-8-localdate-jackson-format> 🡨 Parse LocaldateTime

<https://github.com/larseknu/programmering2_2020/blob/master/Forelesning16_Uke10_FilSkriving/03_FilSkriving_JSON/src/main/java/no/hiof/larseknu/filskriving/Main.java> --> med justeringer med tanke på Oblig 5 🡪 Json skriv og les metode.