1)

Klassene skal samarbeide i den grad at det er loose coupling og high cohesion. Det vil si at hver klasse skal kunne forandres, uten at det vil ha noen endring på klassene som benytter seg av andre klasser. PropertyRegister-klassen vil ha en-mange relasjon med Property-klassen fordi den benytter seg av flere Property-objekter når den blir implementert. me.folk.ntnu.candidatenumber.project.App-klassen vil på den andre siden ha en-en relasjon med PropertyRegister klassen, da denne appen bare har et register "per" objekt av klassen. Det er ingen direkte kobling mellom app-klassen og propertyklassen, den bare bruker propertyklasse. I tillegg er det bare app-klassen som bruker inputHandler. Dette er grunnen til at jeg har valgt den komposisjonen jeg har.

5)

Kopling handler om hvilken grad komponenter er avhengig av hverandre for at implementeringen skal være suksessfull. Min kode er lagd slik at dersom noe i app-klassen skal endres, er det ingenting i de andre klassene som blir berørt da koplingen er løs. Hvis man senere skal endre/vedlikeholde et system, vil det ved høy kopling være stor sannsynlighet for å måtte endre flere moduler dersom en skal blir endret. Innkapsling er en måte a forbedre kopling. Innkapsling, eller å skjule data, separerer gjennomføringen fra «interface», og skjuler da detaljene (tilstandsinformasjon og interne prosesser) fra implementeringen. Dette er blitt gjort i min kode da de andre klassene kun benytter seg av metodene. Som for eksempel «listAllPropertiesMenu» i App-klassen få returnet en String fra register-klassen, som bruker en metode fra property-klassen. Utennoen kunnskap i app-klassen om at dette skjer. Er dermed ikke avhengig av denne informasjonen (løs kopling). Målet med koden er å ha god innkapsling, slik at systemet blir lettere å teste og endre, fordi ingenting annet blir påvirket av endringen enn den modulen/delen som endres.

Cohesion handler om til hvilken grad en enkelt modul (metode, klasse) har en eneste hensikt eller funksjon. Høy cohesion er målet og dette har jeg oppnådd ved å ha klasser med en enste oppgave, som for eksempel InputHandler som bare tar seg av valideringer. Her har jeg brukt abstrahering ved å kun lage metoder og klasser som har en eneste hensikt, ved da å ignorere detaljer som kan bli tatt hånd om av en annen metode/klasse. Dette gjør det også enklere å gjenbruke moduler, fordi de bare har en hensikt/funksjon. Abstraksjon er en måte å oppnå bedre/høyere cohesion i modulene i systemet.