

Rapport for Arbeidskrav 1 – Refaktorering

Fremgangsmåte

Vi startet med å sette oss ned og analysere koden for å få en forståelse av hvordan den fungerte og hvilke endringer vi måtte gjøre, mens vi gjorde dette endret vi også et par variabelnavn for å øke lesbarhet. Etter at vi hadde fått en grei oversikt over koden lagde vi et klassediagram hvor vi brukte prinsippene i GRASP til å utforme hvordan vi tenkte den ferdige refaktorerte koden skulle se ut, klassediagrammet ligger vedlagt i innleveringen. Vi prioriterte å splitte opp koden i klasser og metoder for å gjøre den mer ryddig, her fokuserte vi på å skape low coupling og å ha high cohesion. Når vi hadde kartlagt hvordan vi ønsket å endre koden, fordelte vi arbeidsoppgavene så vi hadde oversikt på hvem som skulle gjøre hva. Vi jobbet hovedsakelig sammen i øvingstimer men vi jobbet også litt individuelt hjemme. For å samhandle ordentlig og passe på at vi kodet i samsvar med hverandre brukte vi git.

Hvordan det gikk

Refaktoreringen av koden gikk veldig greit. Vi kom fort i gang og fikk fordelt koden i nye klasser og metoder, men la raskt merke til at koden vi endte opp med ikke sto helt i samsvar med klassediagrammet vi opprinnelig hadde tenkt oss, så det ble noen endringer på klassediagrammet underveis under refaktoreringen av koden. Vi glemte å lagre første utkastet av klassediagrammet og fortsatte å gjøre endringer på det underveis i prosessen, så vi har kun klassediagram av det ferdige produktet.

Konklusjon

Vi mener selv at koden kunne blitt skrevet bedre, og vi kan se at det er et forbedringspotensial. Allikevel er vi fornøyde med den ferdig refaktorerte koden ut fra vår kompetanse da vi ikke er veldig erfarne med C#. Koden er mye lettere leselig enn den opprinnelige var og vi har hatt fokus på design prinsipper i GRASP, som High Cohesion, Low Coupling, Controller og Information Expert.