

### 17.1

Kirjoitin määrittelydokumentin loppuun ja suunnittelin sen avulla projektia. Toteutettavat toiminnot, käytettävät algoritmit ja tavoiteltavat aikavaativuudet ovat nyt tiedossa. Toteutin myös ensimmäisen toiminnon, matriisien yhteenlaskun. Skalaarikertolaskua toteuttaessa haasteeksi muodostui jo olemassa olevan metodin jäykkyys. Joutunen refaktoroimaan funktion make-matrix niin, että se toimii myös erilaisilla parametreilla.

### 18.1

Viimeistelin skalaarikertolaskun juuri muuttamalla aiemman koodin rakennetta. Tein uusille funktioille testit. Voisin tällä viikolla ottaa selvää Clojure:n yksikkötesteille suunnatuista testikattavuuden raportointi työkaluista (kuten JUnit:illa esim Cobertura).

### 21.1

Toteutin traspoosi-funktion ja kirjoitin sille testit. Liitin projektiini liitännäisen Cloverage (<https://github.com/lshift/cloverage>), joka tuottaa yksikkötestien koodikattavuudesta raportteja. Testini kattoivat 95 % koodiriveistä ja parilla testillä sain ne kattamaan kaiken muun, paitsi pääfunktion/päämetodin. Pääfunktio ei tee mitään, mutta voisin tehdä sillekin testit. Clojure-koodiani alkaa olla yli 60 riviä. Voisin ensi viikolla ottaa selvää, miten Clojure-ohjelmien koodi tulisi asetella. Olio-ohjelmoinnista toimintoja jaetaan metodeihin ja luokkiin single-responsibility periaatetta, riippuvuuksien minimointia ja selkeää rakennetta silmällä pitäen. En ole varma, mitenedellä mainittujaa sovelletaan funktionaaliseen ohjelmointiin.

### 22.1

Aloitin matriisikertolasku-toiminnallisuuden toteuttamisen. Käytin apuna Cormenin kirjan pseudokoodia, jonka soveltaminen Clojurelle/funktionaaliseen ohjelmointikielelle osoittautui haastavaksi. Otin avuksi Clojuren debuggeri-työkalun (<https://github.com/clojure/tools.trace>). Toiminnallisuus jäi osittain kesken ja jatkan sen toteutusta ensi viikolla. Testit eivät myöskään kata uusia funktioita; korjaan sen pian.

Tällä viikolla ohjelman tuleva rakenne ja toteutus selkiytyivät paljon. Käytettävät algoritmit ovat tiedossa ja matriisien käsittelyssä tarvittava teoria on palautunut mieleen. Toteutin myös ensimmäisiä toiminnallisuuksia ja työrutiini pääsi muodostumaan.

Seuraavalla viikolla pyrin toteuttamaan tänään kesken jääneet toiminnot valmiiksi ja uutena toimintona toteutan determinantin laskemisen. Tämän jälkeen joko laajennan toiminnallisuuksia esimerkiksi käänteismatriisin laskemiseen tai keskityn olemassa olevan koodin parantamiseen ja selkeyttämiseen.

Sain kurssikaveriltani (joka ei osaa Clojurea) palautetta, ettei ohjelmani ole luettava. Pyrin tutustumaan Clojuren tyylioppaisiin, jotta koodini olisi luettavaa. Myös funktioiden ja testien jakamista eri tiedostoihin on tutkittava.