



---

# ASCII-GRAFIKKAA

---

Harjoitustyö II, Lausekielinen Ohjelmointi II 2016



Kim Saren
-----------

Tampereen yliopisto Informaatitieteiden yksikkö Matematiikka, 415405 saren.kim.A@student.uta.fi
--

Tein java-koodia käyttäen ASCII-grafiikkaa tulkitsevan ASCIIArt-ohjelman. Ohjelma lukee käyttäjältä syötteen komentoriviparametrina ja koittaa etsiä kyseistä tiedostoa sekä tallentaa sen pohjalta merkkitaulukon muistiin. Mikäli tiedostoa ei löydy, loppuu ohjelman toiminta heti alkuunsa. Ohjelma tulostaa otsikkonsa tasan kerran siinäkin tilanteessa, että komentoriviparametri on virheellinen. Merkkitaulukoita tulkitaan, mikäli ne noudattavat annettuja harmaasävykoodeja.

Mikäli alusta päästään eteenpäin, luodaan tiedoston pohjalta merkki- ja kirjaintaulukko muistiin ja siirrytään ohjelman valintalohkoon. Valintalohko on tehty Dow while-silmukalla, ja jatkaa toimintaansa kunnes käyttäjä antaa "quit"-kehotteen. Valintasilmukassa käyttäjällä on mahdollisuus tulostaa merkkitaulukko tai lukutaulukko, kysellä tietoa taulukosta sekä merkeistä info-komennolla ja suodattaa elementtejä pois "filter [n]"-komennolla. Jos käyttäjä haluaa ladata alkuperäisen kuvan uudestaan, onnistuu se "reset"-komennolla. Silmukka lopetetaan lippumuuttujan vaihdoksella, kun käyttäjä valitsee "quit"-komennon.

Itse ohjelma on jakautunut useisiin eri operaatioihin. Jo pääoperaatiossa kutsutaan erilaisia laskuri- ja luontiominaisuuksia. Varsinainen ohjelman suorittaminen tapahtuu valintaoperaatiossa, jossa jokaiselle komennolla on omia kutsu- sekä apuoperaatioita. Pyrin tekemään jokaisen operaation "järkevyyden rajoissa" niin, että pisimmät operaatiot ovat alle sivun mittaisia ilman kommentointia.

Kokonaisuudessaan toinen harkkatyö sujui mielestäni suhteellisen mukavasti. En sitten tiedä onko kyse siitä, että ohjelmointikokemusta ja -taitoa on kertynyt kurssin aikana enemmän, mutta kurssin ensimmäinen harkkatyö tuntui huomattavasti haastavammalta vuoden ohjelmointitauon jälkeen. Opin työn aikana mm. siirtämään parametreja tehokkaasti operaatiosta toiseen, sekä ylipäätään jäsentämään ohjelmaa loogisen mittaisiin pätkiin operaatioiden avulla. Erityisen hyödyllistä on saman operaation kierrätys, mikäli jokin toinen operaatio vaatii aiemman kaltaisia sovellutuksia. Kokonaisuudessaan käytin tähän harjoitustyöhön n. 12 tuntia.