

Sokkelon ratkaisu oikean käden säännöllä



Rakenne ja toiminta

Maze-ohjelma käyttää kolmea luokkaa kolmessa tiedostossa: ajuriluokka Maze.java sekä apuluokat Generator.java ja In.java. Apuluokkia ei ole selostettu, koska dokumentti koskee vain Maze-luokkaa.

Ohjelma alkaa siten, että ohjelmaa käynnistettäessä syötetään kolme komentoriviparametria. Ensimmäinen on siemenluku, toinen on sokkelon rivien ja kolmas on sarakkeiden lukumäärä. Kaikkien kolmen on oltava kokonaislukuja ja parametreja on annettava 3kpl. Siemenluvulla ei ole itse luvun suhteen rajoitteita, mutta toinen ja kolmas on oltava arvoltaan vähintään 3 ja pariton. Ohjelma päättyy välittömästi jos nämä ehdot eivät täyty.

Parametrien hyväksynnän jälkeen ne välitetään Generator-luokalle, joka parametrien perusteella luo ja palauttaa sokkelon. Sokkelo luomisen jälkeen edetään vaiheeseen, jossa kysytään In-luokan avulla käyttäjän haluamaa toimintoa. Toimintovaihtoehdot ovat liiku (m/move), ratkaise (s/solve) ja lopeta (q/quit). Komento "lopetä" päättää ohjelman välittömästi tervehdysten kera. Komennon luku-metodi sisältää silmukan, joka pyörittää suurinta osaa ohjelman toiminnasta.

Komennolla "liiku" suoritetaan seuraavat toiminnot numerojärjestyksessä:

1. Tarkistetaan sen hetkinen suunta.
2. Siirrytään suuntaa vastaavaan metodiin, joka päättelee mahdolliset seuraavat liikkeet.
3. Liikutaan 1. mahdolliseen suuntaan ja päivitetään uusi sijainti paikkatietotaulukkoon.
4. Tarkistetaan, onko saavuttu uloskäynnille.
5. Kyllä: tulostetaan taulukko, ohjelma päättyy tervehdykseen.
Ei: tulostetaan taulukko, kysytään seuraava komento.

Komento "ratkaise" toimii samalla kaavalla, mutta silmukan suoritus siirtyy toiseen silmukkaan, missä ei kysytä komentoja, vaan sokkelon ratkaisu jatkuu kunnes saavutaan uloskäynnille. Ratkaisu-silmukassa sokkelo tulostetaan vasta kun sokkelo on ratkaistu.

Oma kommentti

Työhön kului aikaa noin 20–24 tuntia.

Tässä harjoitustyössä hyvää/miellyttävää olivat selkeät virstanpylväät ohjelman rakentamisessa. Muutama näistä pylväistä (itselläni ainakin) oli pääsilmukan oikea toiminta, automaattisen ratkaisun silmukka ja ohjelman eri päätöstilanteiden oikea toiminta.

Tässä työssä on selkeät kohdat, jotka ovat kaikille ongelmallisia, mutta en sanoisi työn olleen varsinaisesti vaikea. Kaikki ratkaisut kyllä keksi, kun vain mietti rauhassa tai vaikka hahmotteli ensin paperille. Jos 1 on helpoin ja 10 vaikein, ohjelmoinnin untuvikolle (kuten minä) tämä harjoitustyö oli tasolla 6,5.

Tästä työstä teki ensimmäiseen verrattuna huomattavasti mukavamman se, että aloin tekemään 2 viikkoa ennen palautuksen takarajaa, mikä on huomattavasti aiemmin kuin ekassa työssä (2 päivää ennen palautusta). Ehkä nyt siis osaisin aloittaa työt ajoissa. Voisin myös sanoa, että ajattelukin oli helpompaa kun ei tarvinnut stressata palautusta.

Työstä oli tekijälle hyötyä juurikin em. syystä, mutta myös opin vuoksi. Tässä työssä oppi ja tuli kerrattua paljon metodien välisestä tiedonvälityksestä, parametreista ja metodien kutsuista. Itse opin lisää siitä kuinka ohjelman suoritus tosiaan palaa aina metodin kutsupaikkaan. Tässä työssä oppi myös kommentointia (ainakin jos sen teki hyvin), koska ilman kommentteja tämäkin ohjelma olisi todella sekava muille lukijoille.