# Olio-ohjelmoinnin perusteet 2017

# Harjoitustyö

Komentotulkki

# Toni Kuikka

Tampereen yliopisto

Luonnontieteiden tiedekunta

Matematiikka ja tilastotiede

421568

Kuikka.Toni.M@student.uta.fi

Tulkki

+ muodostaHakemistopolku(Hakemisto) : String

+ luoHakemisto(String)

+ vaihdaHakemistoa(String)

+ luoTiedosto(String)

+ nimeaUudelleen(String)

+ listaaHakemisto(String)

+ kopioiTiedosto(String)

+ poistaTieto(String)

+ listaaRekursiivisesti(Hakemisto)

+ lopetaOhjelma() : Boolean

logiikka

Oope2017HT

+ main(String[])

In

- READER: Scanner

- BARF: String

- AARGH: String

- initializeREADER() : Scanner

- doNotSoGracefulExit(Exception)

+ readInt() : Integer

+ readDouble() : Double

+ readChar() : Character

+ readString() : String

apulaiset

*Tieto*

- nimi: StringBuilder

+ toString() : String

+ compareTo(Tieto) : Integer

+ equals(Object) : Boolean

tiedot

OmaLista

+ hae(Object) : Object

+ lisaa(Object) : boolean

+ poista(Object) : Object

Kayttoliittyma

+ LUOHAKEMISTO: String = ”md”

+ VAIHDAHAKEMISTOA: String = ”cd”

+ LUOTIEDOSTO: String = ”mf”

+ NIMEAUUDELLEEN: String = mv”

+ LISTAAHAKEMISTO: String = ”ls”

+ KOPIOITIEDOSTO: String = ”cp”

+ POISTATIETO: String = ”rm”

+ LISTAAREKURSIIVISESTI: String = ”find”

+ LOPETAOHJELMA: String = ”exit”

+ suoritaohjelma()

oope2017ht

Komennettava

1 1

tulkki

Ooperoiva

0..1

seuraava

0..1

1

0..\*

Lista

omalista

sisältö

1

0..1

ylihakemisto

1

0..\*

käsiteltävä hakemisto

1..2

Tiedosto

- koko: Integer

+ toString() : String

Hakemisto

+ hae(String) : Tieto

+ lisaa(Tieto) : Boolean

+ poista(String) : Tieto

+ toString() : String

fi.uta.csjola.oope.lista

Solmu

- alkio: Object

+ toString() : String

LinkitettyLista

- haeSolmu(Integer) : Solmu

+ lisaaAlkuun(Object)

+ lisaaLoppuun(Object)

+ lisaa(Integer, Object) : Boolean

+ poistaAlusta() : Object

+ poistaLopusta() : Object

+ poista(Integer) : Object

+ alkio(Integer) : Object

+ korvaa(Integer, Object) : Object

*AbstraktiLista*

# koko: Integer

# paikkaOK(Integer) : Boolean

+ lisaaAlkuun(Object)

+ lisaaLoppuun(Object)

+ lisaa(Integer, Object) : Boolean

+ poistaAlusta() : Object

+ poistaLopusta() : Object

+ poista(Integer) : Object

+ alkio(Integer) : Object

+ korvaa(Integer, Object) : Object

+ koko() : Integer

+ onkoTyhja() : Boolean

Luokkakaavio

Oope2017ht-pakkauksessa oleva Oope2017HT-ajoluokka hyödyntää logiikka-pakkauksessa olevan Kayttoliittyma-luokan käyttäjän kanssa vuorovaikuttavaa metodia ohjelman käynnistyessä. Käyttöliittymällä on Tulkki-attribuutti, joka vastaa käyttäjän antamien komentojen varsinaisesta toteuttamisesta käsittelemällä puolestaan omina attribuutteinaan olevia tiedot-pakkauksessa olevia hakemistoja. Hakemisto toteuttaa apulaiset-pakkauksen Komennettava-rajapinnan ja periytyy yhdessä Tiedosto-luokan kanssa Tieto-luokasta. Hakemiston attribuutteja ovat ylihakemisto sekä OmaLista, joka säilöö sen sisällön. OmaLista periytyy fi.uta.csjola.oope.lista-pakkauksen LinkitettyLista-luokasta ja toteuttaa apulaiset-pakkauksen Ooperoiva-rajapinnan.

Omia ajatuksia

Harjoitustyön tekeminen osoittautui erittäin hyödylliseksi, sillä vaikka kurssin harjoitustehtäviä tekemällä oli oppinut hyvin olio-ohjelmoinnin perusteita, sen käytännön merkitys oli jäänyt hieman hämäräksi. Kenties juuri sen takia työn haastavin osuus olikin alkuun pääseminen, kun luokkien välisten suhteiden hahmottaminen oli vaativaa sekä oli vaikeuksia päättää, mistä aloittaa. Kuitenkin hyvällä suunnittelulla sekä esimerkiksi luokkakaavion hahmottelulla pääsi vähitellen alkuun, jonka jälkeen työn tekeminen muuttui helpommaksi.

Erityisesti työ opetti hahmottamaan käyttöliittymän, logiikkaluokan ja tietojen muodostaman kokonaisuuden sekä sen, minkätyyppisiin ohjelmiin olio-ohjelmointia voisi käytännössä soveltaa. Luokkien välisistä assosiaatioista oli ollut kokemusta vain parin harjoitustehtävän verran, mutta harjoitustyön myötä niiden merkityksen ymmärsi täydellisesti. Myös linkitetty lista sekä sen hyödyllisyys tulivat viimeistään tässä vaiheessa tunnetuiksi, ja koodausrutiinia tuli huomattavasti lisää.

Alussa työn tekemiseen kului melko paljon aikaa, ja välipalautuksen lähestyvä deadline oli aiheuttaa suurta stressiä, mutta sen jälkeen tilanne kääntyikin niin päin, että koko ohjelman sai valmiiksi hyvissä ajoin ennen lopullista määräaikaa. Kun luokkahierarkian ja kaikkien luokkien väliset suhteet oli saanut tehtyä, koodaaminen muuttui melko suoraviivaiseksi tehtävänannon noudattamiseksi. Ainoastaan dokumentointi uusilla tavoilla vaati runsaasti opettelua ja siten aikaa. Kaiken kaikkiaan työn aihe tuntui mielenkiintoiselta, ja sen tekeminen oli ylipäänsä erittäin palkitseva kokemus.