

Olio-ohjelmoinnin perusteet II 2019
Harjoitustyö

Simple Oope Shell

Roni-Markus Lumpo

Tampereen yliopisto
Informaatioteknologian ja
viestinnän tiedekunta
Tietojenkäsittelytieteet
427692
roni.lumpo@tuni.fi

Olio-ohjelmointi II:n harjoitustyössä on tehty ohjelma, joka kantaa nimeä Simple Oope Shell, lyhyemmin SOS. Ohjelma koostuu luokista, joiden tiedot muodostavat komentorivi-ikkunaa mukailevan ”käyttöliittymän”. Harjoitustyo-nimisessä pakkauksessa on sisällä neljä kansiota, jotka ovat: apulaiset, iteraattorit, omalista, sekä tiedot. Harjoitustyo-pakkaus sisältää myös käyttöliittymäluokan, tulkkiluokan sekä Oope2HT-luokan, jonka avulla SOS käynnistetään.

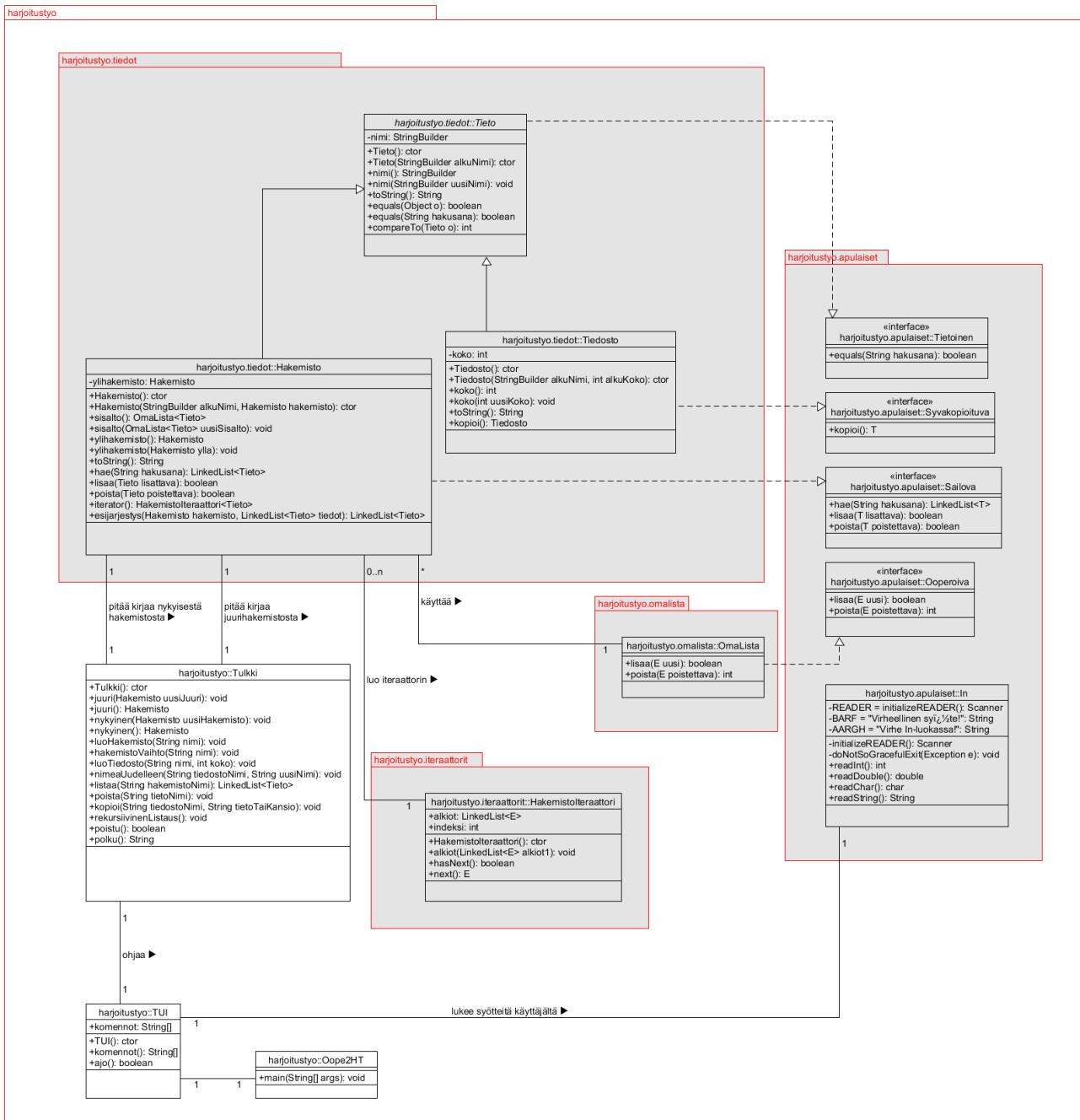
Apulaiset-kansio sisältää opettajan antamat rajapinnat ja luokat In, Ooperoiva, Sailova, Syvakopioituva ja Tietoinen. Iteraattorit-kansio sisältää hakemistoiteraattoriluokan. Omalista-kansio sisältää LinkedList-luokasta perityn omalistaluokan ja tiedot-kansio sisältää tietoluokan ja siitä periytyvät hakemistoluokan, tiedostoluokan.

Käyttöliittymä avataan suorittamalla Oope2HT-luokan main-metodi. Main-metodissa luodaan käyttöliittymäolio sekä kutsutaan käyttöliittymäolion metodia, joka sisältää ohjelman pääsilman. Tulkki-olio luodaan käyttöliittymäluokan rakentajassa. Pääsilman tulostetaan tervehdysteksti käyttäjälle, ja kysytään kommentia. Käyttäjä syöttämät komennot luetaan näppäimistöltä harjoitustyo.apulaiset.In-luokalla, jonka jälkeen käyttöliittymäluokka pilkkoo komennon sekä parametrit osiin, mikäli parametreja on.

Pilkkomisen jälkeen, jos komento ja parametrit ovat oikeellisia, käyttöliittymäluokasta kutsutaan tulkkiluokan operaatiota, joka on komennolle sopiva. Komennon oikeellisuus tarkistetaan käyttöliittymäluokassa ja mikäli komento tai parametri(t) ovat virheellisiä, käyttäjälle tulostetaan virheilmoitus ja luetaan käyttäjältä syöte uudestaan. Tulkkiluokka pitää huolen käyttöliittymän muodostavien olioiden kommunikoinnista, ja kutsuukin näiden operaatioita komentojen mukaisesti.

Komentoja ovat:

- md = kansion tekeminen, tulkkiluokka luo uuden hakemisto-olion ja lisää sen nykyhakemiston omalistalle.
- cd = hakemiston vaihtaminen välittömään yli- tai alikansioon tai suoraan juurihakemistoon, vaihtaa tulkkiluokan nykyinen-attribuutin arvoksi parametrina saadun välittömän yli- tai alikansion tai parametrittömanä asettaa juurikansion nykyiseksi kansioksi.
- mf = tiedoston tekeminen, tulkkiluokka luo uuden tiedosto-olion ja lisää sen nykyhakemiston omalistalle.
- mv = tiedoston nimen muuttaminen, hakee parametrina saadun tiedosto-olion nykyhakemistosta ja käyttää löydetyn tiedosto-olion setteriä uudelleennimeämisessä.
- ls = listaus, esim. koko kansion sisällön listaus tai tiettyjen tiedostojen listaus, tulkkiluokka luo LinkedListin, johon on koottu kaikki parametrina saadut oliot ja käyttöliittymäluokka tulostaa tulkkiluokan palauttaman LinkedListin.
- rm = poisto, tulkkiluokka hakee ja poistaa parametrina saadut oliot nykyhakemistosta.
- cp = tiedoston kopiointi uudella nimellä, tiedoston kopiointi välittömään yli- tai alikansioon samalla nimellä tai useiden tiedostojen kopiointi välittömään yli- tai alikansioon samalla nimellä,
- find = käyttöliittymän kaikkien kansioiden ja niiden sisältöjen listaus, tulkkiluokka luo uuden iteraattorin ja tulostaa alkioita niin kauan, kunnes iteraattorin hasNext()-operaatio palauttaa false.
- exit = käyttöliittymän sammutus, tulkkiluokka palauttaa totuusarvon false, joka pysäyttää käyttöliittymäluokassa suoritettavan pääsilman. Käyttöliittymäluokassa tulostetaan pääsilman jälkeen käyttäjälle tieto siitä, että käyttöliittymä on sammutettu.



Kuva 1. UML-kaavio harjoitustyo-pakkauksesta.

Kuvassa näkyvän kaavion punaiset ulkoreunat kuvastavat harjoitustyo-pakkausta. Harjoitustyo-pakkauksen sisällä on neljä erillistä kansiota: harjoitustyo.tiedot, harjoitustyo.omalista, harjoitustyo.apulaiset ja harjoitustyo.iteraattorit. Harjoitustyo-pakkauksen sisällä on myös Tulkki-luokka, TUI-luokka sekä ajoluokka Oope2HT.

Abstrakti Tieto-luokka toteuttaa rajapinnan Tietoinen. Hakemisto-luokka ja Tiedosto-luokka periytyvät Tieto-luokasta. Tiedosto-luokka toteuttaa rajapinnan Syvakopioituva ja Hakemisto-luokka toteuttaa rajapinnan Sailova. OmaLista-luokka toteuttaa rajapinnan Ooperoiva ja yksi OmaLista-luokka voi säilöä yhden tai useamman Hakemisto-luokan olion sisältöä ja tietoja. OmaLista-luokka voi myös säilöä yhden tai useamman Tiedosto-luokan olion tietoja.

HakemistoIteraattori-luokan olio luodaan Hakemisto-luokan iterator()-operaatiossa. Kyseisen operaation palauttamaa iteraattoria käytetään Tulkki-luokan rekursiivinenListaus()-operaatiossa. Yksi Tulkki-luokan olio voi ohjata yhtä tai useampaa Hakemisto-luokan oliota kutsumalla niiden operaatioita.

Yhtä Tulkki-luokan oliota ohjaa yksi TUI-luokan olio. TUI-luokan olio kysyy käyttäjältä syötteitä yhden In-luokan olion avulla. Oope2HT-luokka sisältää main-metodin, jossa luodaan TUI-luokan olio.

Työ oli mielestäni melko haastava, mutta koin sen enemmän positiivisena haasteena, kuin negatiivisena. Työhön meni runsaasti aikaa, noin 35h yhteensä. Opin harjoitustyön avulla olio-ohjelmoinnin perusteita ja uskonkin, että tästä eteenpäin on helpompaa hahmottaa olioperustaisen ohjelman rakennetta.

Harjoitustyössä haastavinta oli eri olioiden operaatioiden ketjuttaminen niin, että parametrit vastaavat toisiaan ja myöskin, että jos yksi operaatio kutsuu ketjussa montaa muuta operaatiota, niin miten päästään käsiksi itse lopputuotteeseen. Virheenkorjaus oli myös haastavaa, sillä monessa operaatiossa ja monessa eri luokassa saattoi olla virheitä. Näiden kaikkien virheiden kiinni saaminen vei ehkä eniten aikaa yksittäisistä tehtävistä harjoitustyössä.