

Suunnitteludokumentti

ilpo-ryhmä

Helsinki 6.4.2005

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Arttu Eloranta
Risto Kuusterä
Antti Pulkkinen
Tommi Rajala
Tuomo Saarinen
Jari Suominen

Ohjaaja

Topias Marttila

Asiakas

Anni Rytönen

Vastuhenkilö

Juha Taina

Tekniset asiantuntijat

Juhani Haavisto
Harri Laine
Turjo Tuohiniemi

Kotisivu

<http://www.cs.helsinki.fi/group/ilpo>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	25.3.2005	Ensimmäinen versio
1.0	31.3.2005	Technical review:ssä käsiteltävä lopullinen versio
1.0.1	4.4.2005	Osittain korjattu versio
1.0.2	6.4.2005	Jäädettäväksi tarjottava versio

Sisältö

1 Johdanto	1
1.1 Taustaa	1
1.2 Yleisarkkitehtuuri	1
1.3 Järjestelmän tekninen kuvaus	2
1.4 Termit ja määritelmät	2
2 Arkkitehtuurin kuvaus	4
2.1 Sisäiset rajapinnat	4
2.1.1 Kurssi.properties-tiedosto	4
2.1.2 Yleinen.properties-tiedosto	5
2.1.3 HTML-kirjoittaja	6
2.1.4 Keskeneneräisten ryhmittelyiden käsittelijä	6
2.2 Ulkoiset rajapinnat	6
2.2.1 Desmond	6
2.2.2 Ilmo	7
2.3 Opiskelijanumerotulostin	8
3 Luokkarakenne	8
3.1 Model	8
3.2 Control	8
3.3 View	8
3.4 IO ja DB	10
4 Käyttöliittymä	10

Liitteet

1 Sekvenssikaaviot

2 Kurssi.properties-tiedosto

3 Yleinen.properties-tiedosto

4 Javadoc

1 Johdanto

Joillain kursseilla, kuten Ohjelmistotuotantoprojekti, Tieteellinen kirjoittaminen ja Opettajatuutorointi, opiskelijat jaetaan pienryhmiin ilmoittautumisten yhteydessä annettujen taustatietojen ja ryhmätoiveiden mukaan. Tavoitteena ryhmäjaossa on toisaalta muodostaa mahdollisimman tarkoituksenmukaisia, tasakokoisia ja/tai täysiä ryhmiä, ja toisaalta täyttää opiskelijoiden esittämiä toiveita niin paljon kuin mahdollista. Tarkoituksenmukaisuus voi tarkoittaa kurssista riippuen erilaisia asioita. Esimerkiksi tieteellisen kirjoittamisen kurssilla samaan ryhmään kuuluvien opiskelijoiden erikoistumisalueiden tulisi olla yhteneviä. Toisaalta ohjelmistotuotantoprojektissa samaan ryhmään on valittava eri tavoin suuntautuneita henkilöitä. Tällä hetkellä vastuuhenkilö tekee tämän jaottelun käsin paperilla. Tarkoitus on tuottaa ohjelmisto, joka helpottaa jaottelijan tehtävää. Suunnittelu-dokumentti esittää tulevan ohjelmiston, jota ollaan tuottamassa ryhmiinjaottelijan avuksi. Siinä kuvataan ohjelmiston rakennetta ja suunnittelun pohjaksi aiottua luokkarakennetta.

1.1 Taustaa

Kursseilla Ohjelmistotuotantoprojekti, Tieteellinen kirjoittaminen ja Opettajatuutorointi on keskenään erilaiset sisällölliset ja hallinnolliset vaatimukset. Esimerkiksi Ohjelmistotuotantoprojektissa ryhmän koon on oltava 4–6 opiskelijaa. Kurssilla on pakolliset esitietovaatimukset, ja joissain aiheissa saattaa olla lisävaatimuksiakin suoritetuista esitietokursseista. Opettajatuutorointiin ilmoittautuvat opiskelijat ovat eri vaiheissa opinnoissaan ja suorittavat niitä eri tahtiin. Osa opiskelijoista tietää, mihin haluaa suuntautua ja suuntautumiset vaihtelevat. Ryhmät pyritään muodostamaan siten, että opiskelu olisi opiskelijoille mielekästä.

Ilpo on Helsingin yliopiston tietojenkäsittelytieteen laitoksella toteutettava projekti. Projekti toteutetaan osana opintojaksoa Ohjelmistotuotantoprojekti. Projektin tarkoitus on tuottaa sovellus, joka auttaa opintojaksojen Opettajatuutorointi, Tieteellinen kirjoittaminen ja Ohjelmistotuotantoprojekti vastuuhenkilöitä ryhmäjakojen teossa. Kaikki projektin aikana tuotettava materiaali tulee saataville ryhmän kotisivuilla:

<http://www.cs.helsinki.fi/group/home/ilpo>

1.2 Yleisarkkitehtuuri

Ohjelmisto on jaoteltu viiteen osaan. Nämä osat ovat Model, View, Control, IO ja DB. Syy jaotteluun on ohjelmiston helpompi käsiteltävyys varsinkin päivitystilanteissa. Samaa jaottelua käytetään tässä dokumentissa, jotta lukijalle tulisi selkeämpi kuva tuotettavasta ohjelmistosta. Model vastaa kaikkia ydinosasia, joilla muodostetaan mm. Opiskelija-oliot ja Kurssi-oliot. Model:n luokilla on vastuullaan kaikki looginen toiminnallisuus. View:n luokilla luodaan käyttäjälle tarjottava käyttöliittymä. Control:n luokat huolehtivat käyttöliittymän toiminnallisuudesta. Haluttaessa käyttää ohjelmaa, Control:n käsittelevät käyttäjän syötteet. DB:n luokat hoitavat ohjelmiston tietokantayhteydet. Sen osasilla parsitaan tietokannoista tuleva tieto ja huolehditaan yhteyksien luomisesta ja sulkemisesta.

IO:n luokat käsittelevät kaikki tulostuspyynnöt. Johtuen ilmo-järjestelmän huonosta rakenteesta, järjestelmän toteuttaminen on monimutkaista. Järjestelmän toteutusta on kuvattu liitteinä olevina sekvenssikaavioina (liite 1).

1.3 Järjestelmän tekninen kuvaus

Ohjelma toimii Java-sovelluksena graafisella käyttöliittymällä terästettynä. Laitevaatimuksena on, että sovellus toimii tavallisessa TKTL:n pöytäkoneessa (ks. termit). Tämä tarkoittaa implisiittisesti sitä, että järjestelmää voidaan käyttää ainoastaan TKTL:n sisäverkosta käsin. Käyttöjärjestelmistä sen verran, että ohjelma toimii ainakin Linuxissa. Windowsissa ohjelma ei välttämättä tule toimimaan. Java-versiona käytetään JDK 5.0:aa (eli J2SE SDK 1.5.0) ja Desmond-tietokantaan ollaan yhteydessä JDBC-rajapinnan kautta. Työn alla olevat ryhmittelyt tallennetaan levyille sarjallistuvina olioina ja valmiit ryhmittelyt Ilmo-tietokantaan jo olemassa olevan PL/SQL-skriptin avulla.

1.4 Termit ja määritelmät

Bodbacka: TKTL:n tietokantapalvelin.

Desmond (<http://ilmo.cs.helsinki.fi/desmond/>): Aiempien ohjelmistotuotantoprojektien tuottama opintojensuunnittelujärjestelmä. Desmondin www-selaimessa toimivalla käyttöliittymällä opiskelijat voivat suunnitella opintojaan lukukausittain. Desmondissa on myös muita toimintoja, jotka auttavat esimerkiksi tuutoroinnissa, mutta ne ovat projektin kannalta epärelevanttejä.

desm_: Etuliite, jota käytetään viitattaessa Desmond-järjestelmän tietokantatauluihin.

Ilmo (<http://ilmo.cs.helsinki.fi/>): Kurssi-ilmoittautumisjärjestelmä opiskelijoille. Järjestelmä on tarkoitettutietojenkäsittelytieteen laitoksen kursseille ilmoittautumiseen. Ilmolla ilmoitaudutaan myös matematiikan kursseille, mutta sillä ei ole tämän projektin kannalta juuri mitään merkitystä.

Iipo: Sovelluksen toteuttavan ohjelmistotuotantoprojektiryhmän ja lopullisen järjestelmän nimi.

JAR (Java Archive file format): JAR:n avulla voidaan pakata useita tiedostoja yhteen JAR-tiedostoon. Tyypillisesti JAR-tiedostosisältää luokkatiedostoja sekä sovelluksiin ja sovelmiin liittyviä lisäresursseja.

JDBC (<http://java.sun.com/products/jdbc/>): Sun Microsystemsin JDBC-teknologia on Java-kielelle toteutettu sovellusrajapinta (API, Application Programming Interface) joka tarjoaa liityntämahdollisuuden useiden eri valmistajien relaatiotietokantoihin ja mm. laskentataulukoihin (spreadsheets).

Järjestyskriteeri: Järjestyskriteerit ovat Ilmo-lomakkeen yksittäisiä vastauksia: esimerkiksi suuntautumislinja, kieli, esitiedot, opiskelijan tila.

Kontti: ATK-osaston tietokantapalvelin.

Kurki: Laitoksen henkilökunnan käyttöön tarkoitettu kurssikirjanpitojärjestelmä johon merkitään mm. opiskelijoiden tekemät laskuharjoitukset. Käyttää samaa tietokantaa kuin Ilmo.

Kurssi: Kurssi on opintojakson ilmentymä. Kurssi määritellään erikseen kyseessä olevalle periodille, lukukaudelle ja -vuodelle.

Kurssi.properties: Kurssi.properties-tiedosto sisältää yhden kurssin asetustiedostot. Tiedostossa on tiedot, joiden avulla kurssi voidaan yksilöidä ja tiedot, joiden avulla ilpo pystyy parsimaan opiskelijoiden vastaukset luettavaan muotoon.

Käyttäjät: Ilpolla ryhmittelyä suorittava henkilö.

Lukitus: Yhden opiskelijan kiinnittäminen pysyvästi tiettyyn ryhmään siten, että opiskelija ei näy enää muissa ryhmissä.

Oodi (<http://www.helsinki.fi/weboodi/>): Opiskelun ja opetuksen tuen tietojärjestelmä. Oodista löytyvät muun muassa opiskelijoiden opintosuoritukset.

Opintojakso: Opintojakso määrittelee opetettavalle aihepiirille nimen ja laajuuden sekä muut yleisetluentokerrasta riippumattomat tiedot.

Opiskelija: Kurssille ilmoittautunut henkilö, jota sijoitetaan ilpolla ryhmiin.

PL/SQL (Procedural Language/SQL): PL/SQL on laajennus SQL-kyselykieleen. Se lisää perus-SQL:ään proseduraalistenohjelmointikielten rakenteita, jolloin tuloksena on perus-SQL:ää ilmaisuvoimaisempi rakenteellinen kieli.

Ryhmittely (Ryhmien jako): Opiskelijoiden jakaminen ryhmiin.

Ryhmä: Kokonaisuus mihin ilpolla siirretään opiskelijoita. Ryhmässä on opiskelijoita ja henkilökunnan jäseniä.

Ryhmälukitus: Ryhmän lukitseminen siten, että kaikki ryhmässä olevat opiskelijat eivät enää näy muissa ryhmissä.

Sarjallistuva olio: Ohjelmointikielen olio, jonka tila voidaan konvertoida tavuvirraksi (ja toisinpäin). Käytännössä tämä tarkoittaa, että koko olio voidaan esimerkiksi tallentaa kiintolevylle myöhempää käyttöä varten. Sarjallistuvia olioita on ainakin Java-ohjelmointikielissä.

Sisäinen tietorakenne: Olorakenne, joka on pelkästään ilpon sisäisessä käytössä ja mihin on tallennettu tietoja.

Tavallinen TKTL:n pöytäkone: Käyttöjärjestelmänä Linux graafisella ikkunointiympäristöllä, näyttö 1280x1024 resoluutiolla sekä vähintään 1.5 GHz suoritin ja vähintään 512Mt keskusmuistia.

TKTL: Helsingin yliopiston Tietojenkäsittelytieteen laitos.

Tulosryhmä: Järjestyskriteerin mukaisesti tehty järjestely jakaa opiskelijat tulosryhmiin.

Yleinen.properties: Yleinen.properties-tiedosto sisältää ohjelman tarvitsemat yleiset asetustiedot. Näitä ovat tietokantapalvelimen osoite ja portti sekä tietokannan nimi.

2 Arkkitehtuurin kuvaus

Tässä luvussa kuvataan sisäisten rajapintojen ja ulkoisten rajapintojen arkkitehtuuria. Ulkoiset järjestelmät ovat meistä riippumattomia, olemassa olevia ohjelmistoja, kuten Desmond ja Ilmo. Niihin päästään käsiksi rajapintojen kautta. Sisäiset rakenteet ovat niitä, jotka luomme ja joilla mm. otetaan rajapintayhteydet ulkoisiin järjestelmiin. Arkkitehtuurin kuvaukseen kuuluu myös kuvaus Opiskelijanumerotulostimesta, jolla päivitetään tietoja Desmondiin. Kaikista kohdista on laajemmin tietoa aliluvuissa.

2.1 Sisäiset rajapinnat

Sisäiset rajapinnat liittyvät tässä yhteydessä ohjelmiston asetuksiin ja kurssi-ilmottautumisten parsimiseen. Tässä käsitellyt Kurssi.properties-tiedosto ja Yleinen.properties-tiedosto ovat vahvasti yhteydessä ulkoisiin rajapintoihin. Niiden tarkempi rakenne kuvataan aliluvuissa.

2.1.1 Kurssi.properties-tiedosto

Kurssi.properties-tiedosto (liite 2) sisältää ohjelman tarvitsemat kurssikohtaiset tiedot. Näitä tietoja ovat: kurssin nimi, kurssikoodi, kurssin järjestysnumero, lukukausi ja lukuvuosi. Tämän lisäksi kurssi.properties-tiedostossa kerrotaan Ilpolle, miten ohjelma parsii yhteen kurssille ilmoittautumisen yhteydessä kysytyt kysymykset ja kurssille ilmoittautuneet opiskelijat. Ilpon parsiessa Ilmosta haetut kysymykset luettavaan muotoon, lukee se kurssi.properties-tiedostosta ohjeet, miten sen tulisi parsia mikäkin kysymys.

Kurssi.properties-tiedosto pitää myös sisällään joukon muuttujia, joilla määritellään ryhmäjaon HTML-tulosteeseen liittyvät tekstit. Näitä ovat esimerkiksi tiedot ryhmien ensimmäisistä kokoontumisajoista ja ohjeet ylivuotoryhmäläisille, sekä kaikki sivulla esiintyvät otsikot. kurssi.properties-tiedoston sisältö on seuraavanlainen:

Kurssin perustiedot

koodi (esim. koodi = 58110-3):

Kertoo kurssin kurssikoodinumeron

kurssijärjestysnumero (esim. kurssiJarjestysnumero = 1):

Kertoo kurssin järjestysnumeron, eli jos sama kurssi luennoidaan useampaan kertaan saman lukukauden aikana, niin tämän numeron avulla erotetaan kurssit toisistaan.

lukukausi (esim. lukukausi = k):

Kertoo käydäänkö kurssi syksyllä (s), keväällä (k) vai kesällä (v).

lukuvuosi (esim. lukuvuosi = 2005):

Kertoo lukuvuoden, jolloin kurssi käydään.

Parsimisessa tarvittavat tiedot Jotta opiskelijoiden vastaukset saataisiin parsittua oikein, täytyy kurssin vastuuhenkilön määrittää kurssi.properties-tiedostoon, että millä tavalla kysymykset parsitaan. Seuraavat parsimisessa tarvittavat tiedot kirjoitetaan muotoon: kysy-

`mys.numero.optio = arvo.`

kannassa (esim. `kysymys.1.kannassa = 3`):

Kertoo Ilpolle, mikä on Ilpon kysymyksen numero 1 numero Ilmon tietokannassa.

preferenssi (esim. `kysymys.2.preferenssi = 1`):

Kertoo Ilpolle, että mikä järjestysnumero annetaan vastauksen osalle. Tätä optiota tarvitaan ilmoittautumisen yhteydessä kysytyissä kysymyksissä, joissa annetaan vastaus avoimeen tekstikenttään ja vastaus pitäisi jakaa useampaan osaan. Esimerkiksi jos opiskelijalta kysytään kiinnostuksen kohteita annetusta listasta ja pyydetään vastaamaan pelkästään pilkuilla erotetuilla numeroilla, niin preferenssi-option avulla saadaan vastauksen kiinnostuksen kohteet eriteltyä omiksi riveikseen. Lisäksi preferenssi-optiota käyttämällä määritellään myös, että kuinka monta vaihtoehtoa opiskelijan vastauksesta huomioidaan.

csv (esim. `kysymys.2.tyyppi = csv`):

Kertoo Ilpolle, että vastauksen tyyppi on pilkuilla erotettu arvo (=csv, comma separated values). Tätä optiota tarvitaan kertomaan Ilpolle, että miten se parsii tekstikenttä-tyyppisiä vastauksia. Ilpossa erotellaan vain pelkistä numeroista koostuvia vastauksia.

vv (esim. `kysymys.2.p1.vv.3 = Käyttöliittymät`):

Kertoo Ilpolle kysymyksen vastausvaihtoehtonumeroa vastaavan tekstin. Tätä tietoa tarvitaan joissain tekstikenttä-tyylisissä kysymyksissä.

html.otsikko (esim. `html.otsikko = Kevään 2005 ohjelmistotuotantoprojektien alustava ryhmäjako`):

Kertoo HTML-parsijalle, mikä on ryhmienjaon lopputuloksena tehtävän html-sivun otsikko.

html.ingressi (esim. `html.ingressi = Seuraavassa on 12.01.2005 päivätty lista kevään 2005 ohjelmistotuotantoprojekteista`):

Kertoo HTML-parsijalle, mikä on sivun sisältöä kuvaava teksti.

html.kokoontumisajat (esim. `html.kokoontumisajat = (24.1.2005 - 28.1.2005). Projektit päättyvät viimeistään viikolla 18 (2.5.2005 - 6.5.2005)`):

Kertoo HTML-parsijalle kokoontumisajat-tekstin.

html.jonotusotsikko (esim. `html.jonotusotsikko = Jonotuslista`):

Kertoo HTML-parsijalle jonotusotsikko-tekstin.

html.jonotusohjeet (esim. `html.jonotusohjeet = Seuraavat henkilöt ovat jonossa: Manta-tarms Suominen, Risto-Ville Kuusinen, Matias Torppila`):

Kertoo HTML-parsijalle jonotusohjeet-tekstin.

html.spotiosoite (esim. `html.spotiosoite = ohtu@cs.helsinki.fi`):

Kertoo HTML-parsijalle sähköpostiosoite-tekstin.

2.1.2 Yleinen.properties-tiedosto

Yleinen.properties-tiedosto sisältää ohjelman tarvitsemat yleiset asetustiedot. Yleisiä tietoja ovat tietokantapalvelimen osoite ja portti sekä tietokannan nimi.

tk.osoite: Tietokantapalvelimen osoite

tk.portti: Tietokantapalvelimen portti

tk.nimi: Tietokannan nimi

Liitteenä (liite 3) esimerkki yleinen.properties-tiedostosta.

2.1.3 HTML-kirjoittaja

HTML-kirjoittaja tulostaa Ilmo-ryhmiin liitetyn ryhmittelyn HTML-sivun osana. Tulostiedosto vastaa osoitteessa <http://www.cs.helsinki.fi/group/ohtu/k-2005/ryhmajako.html> olevaa sivua, kuitenkin siten, että se pitää sisällään ainoastaan *header loppuu-* ja *trailer alkaa-*kommenttien välisen alueen. Tiedosto kirjoitetaan käyttäjän valitsemaan paikkaan ja siinä esiintyvät tekstivakiot määritellään kurssi.properties-tiedostossa. HTMLKirjoitin on yhden metodin kirjastoluokka, joka saa parametrinaan kurssi-olion ja kohdetiedoston nimen. Kirjoituksessa käytetään apuna Tiedostokirjoittaja-luokan metodeja.

2.1.4 Keskeneräisten ryhmittelyiden käsittelijä

Keskeneräisten ryhmien käsittelijällä on kaksi toimintoa: nykyisen ryhmäjaon tallettaminen halutulla nimellä levyille ja valitun ryhmäjaon lataaminen levyiltä. Osajärjestelmän toiminta perustuu olioiden sarjallistuvuuteen: kun sarjallistuva kurssi-olio kirjoitetaan levyille myös kaikki oliot joihin se viittaa kirjoitetaan talteen. Ryhmittelyn latauksen yhteydessä ohjelma palautetaan talletettuun tilaan.

2.2 Ulkoiset rajapinnat

Ulkoisia rajapintoja ovat Desmond ja Ilmo. Molemmat järjestelmät ovat suunniteltu omiin käyttötarkoituksiinsa, joten niiden hyödyntäminen Ilpo-projektissa on hieman kankeaa. Desmondin tietokannan tietoja ei voida päivittää mitenkään Ilposta-käsin. Tämän takia Ilpon tulee pystyä tuottamaan lista niiden opiskelijoiden opiskelijanumeroista, joiden tiedot halutaan päivittää Desmondiin. Tekstimuotoisen tiedoston tulee sisältää yksi opiskelijanumero riviä kohti. Tämän tiedoston avulla Desmondin ylläpitäjä (tai vastaava henkilö) päivittää Desmondin tiedot ajantasaiseksi.

Desmondin palvelun käyttämisestä on kerrottu tarkemmin seuraavassa aliluvussa. Ilmo-tietokannan rakenne ei ole, ainakaan Ilpon näkökulmasta, kovin järkevästi toteutettu. Esimerkiksi opiskelijoiden toiveet kurssijaon suhteen ovat pilkulla erotettuna yhdessä tekstimuotoisessa kentässä. Muun muassa tämän johdosta Ilpo joutuu parsimaan tietoa "tarpeettomasti". Ilmosta on kerrottu tarkemmin luvun toisessa alikappaleessa.

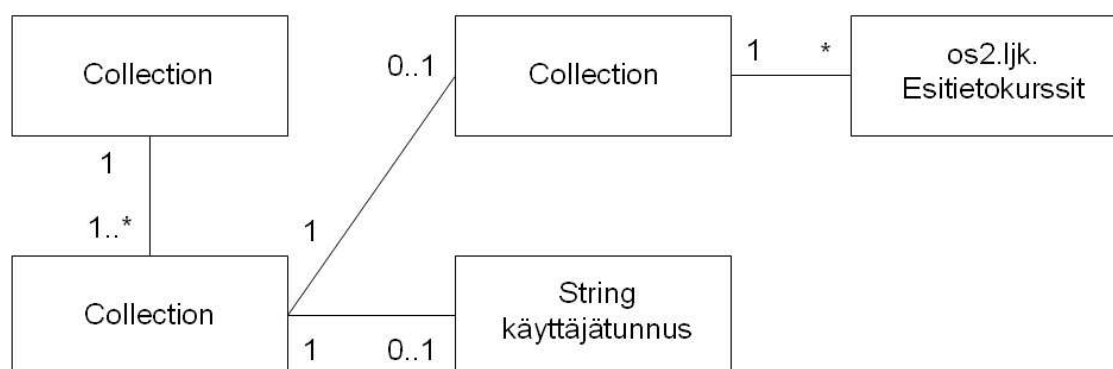
2.2.1 Desmond

Yhteys Desmondiin hoidetaan projektiryhmälle toimitetun Desmond.jar-paketin avulla. Paketti tarvitsee toimiakseen luokkapolon määrittelyn. Esimerkiksi:

java-classpath Desmond.jar:/opt/jdbc/classes12_01.zip: Ohjelma

Ohjelman kääntäminen tapahtuu vastaavasti, parametrit pitää huomioida Ant-tiedostossa. Puuttuvien esitietojen tarkistamiseksi paketista kutsutaan Esitiedot luokan staattista tarkistaEsitiedot-metodia. Metodille annetaan ensimmäisenä parametrina tarkastettavien opiskelijoiden käyttäjätunnukset sisältävä olio, jonka tulee toteuttaa Collection-rajapinta. LinkedList-olio sopii tähän tarkoitukseen hyvin. Toisena parametrina annetaan profiloitavan kurssin kurssikoodi, esim. "581260-4". Metodi palauttaa opiskelijoilta puuttuvat esitiedot Collection-oliona. Palautettu olio sisältää joukon Collection-olioita. Nämä oliot taas sisältävät opiskelijan String-muotoisen käyttäjätunnuksen sekä Collection-olion, joka sisältää puuttuvat kurssit os2.ljk.EsitietoKurssit-olioina. EsitietoKurssit-oliot sisältävät kaksi metodia: annaKurssikoodi() ja annaTaso(). Molemmat näistä palauttavat String-muotoisen arvon ja annaTaso-metodi palauttaa arvonaan joko "pakollinen", "hallittava" tai "suositeltu".

Kuvassa 1 on selvennetty tarkistaEsitiedot-metodin palauttaman Collection-olion rakennetta. Huom! Jos käyttäjätunnus-olio on olemassa, niin silloin myös sen "sisarsolmu" eli Collection on olemassa. Jos sitä ei ole olemassa niin sisarsolmuakaan ei ole olemassa. Tämä pätee myös toisin päin.



Kuva 1: tarkistaEsitiedot-metodin paluuarvon "rakenne"

2.2.2 Ilmo

Ilpo-ryhmän käyttöön on luotu kaksi teoriassa identtistä Oracle-testikantaa, toinen Konttiin ja toinen Bodbackaan. Ryhmä päätti käyttää projektin tuotoksena syntyvässä ohjelmassa ainoastaan Kontti-yhteyttä. Javan JDBC:n avulla toteutettava tietokantayhteys tarvitsee toimiakseen polkumäärittelyn, joka viittaa Oraclen JDBC-tietokanta-ajuriin. Tämä onnistuu esimerkiksi komennolla:

java -classpath /opt/oracle-client-10.1.0/jdbc/lib/ojdbc14.jar: Ohjelma

Huomaa myös Desmond.jar-paketin vaatimukset polkumäärittelyjen suhteen. Tietokannan nimi, osoite ja portti määritellään yleinen.Properties-tiedostossa. Tästä enemmän arkitektuurin kuvaus-luvussa. Valmiiksi ryhmitellyt ryhmät tallennetaan Ilmo-tietokantaan. Tallentaminen tapahtuu joko PL/SQL-skriptin avulla tai sitten suoraan SQL-lauseilla. Ryhmä ei ole saanut riittävästi tietoa tästä asiasta lukuisista pyynnöistä huolimatta.

2.3 Opiskelijanumerotulostin

Opiskelijanumerotulostimella kirjoitetaan tiedostoon niiden opiskelijoiden opiskelijanumerot, joiden tiedot halutaan päivitettävän Desmond-järjestelmään. Käytännössä tämä sisältää profiloitavalle kurssille ilmoittautuneet opiskelijat. Tiedosto sisältää yhden opiskelijanumeron riviä kohti. Luokan ilmentymä saa kirjoitusmetodilleen parametrina Collection-rajapinnan toteuttavan olion, joka sisältää Opiskelija-olioita. Kirjoituksessa käytetään apuna Tiedostokirjoittaja-luokan metodeja.

3 Luokkarakenne

Tässä luvussa kuvataan ohjelmiston luokkarakenne. Luokat on jaoteltu pakkauksiin ja pakkauksia on luotu viisi kappaletta. Luokkaryhmät, joita pakkaukset pitävät sisällään ovat nimeltään Model, Control, View, IO ja DB. Kaikki ryhmät käsitellään seuraavissa aliluvuissa. Tarkempi kuvaus jokaisesta pakkauksesta löytyy API-kirjastosta, joka on liitteenä Javadoc muodossa (liite 4).

3.1 Model

Model-luokat sijaitsevat `fi.helsinki.cs.ilpo` -pakkauksessa. Kyseessä on järjestelmän sisäiseen tietorakenteeseen liittyvät luokat, joissa on myös kaikki niihin liittyvä looginen toiminnallisuus. Kurssi ja Opiskelija luokat ovat ne millä pääsääntöisesti operoidaan. Niillä muodostetaan Kurssi-, Ilporyhmä- ja Opiskelija-olioita. Opiskelija-olioita sijoitellaan Ilporyhmä-olioihin. Muut luokat ovat avusteisia näille luokille. Ne liittyvät kursseihin, opiskelijoihin ja ryhmiin kuuluvien tietojen käsittelyyn. Sisäinen tietorakenne on kuvattu kuvassa 2.

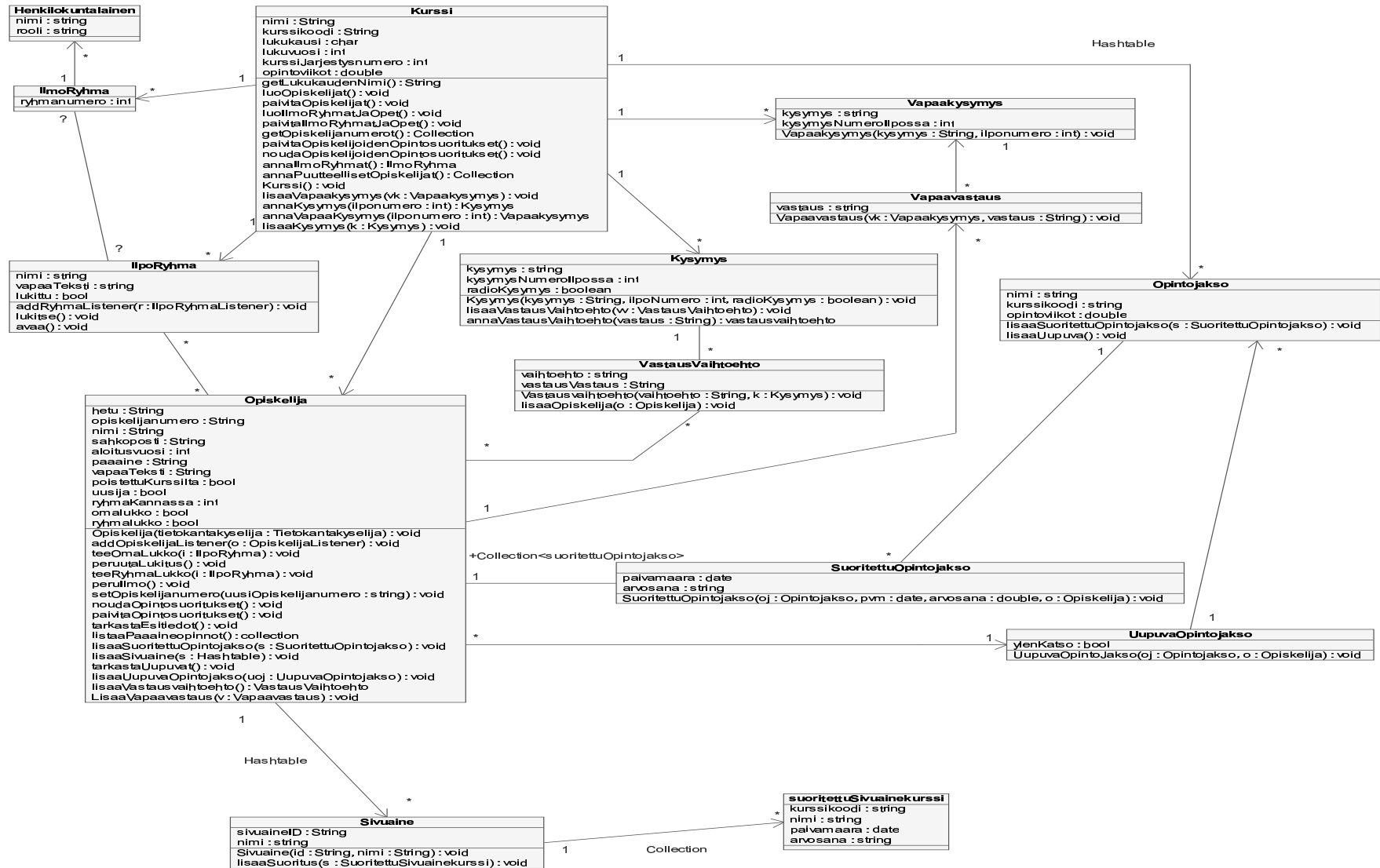
3.2 Control

Control-luokat sijaitsevat `fi.helsinki.cs.ilpo.control` -pakkauksessa. Kyseessä on käyttäjän syötteiden proseissoinnin hoitavat luokat. Suunnitteluvaiheessa eri Control-luokkia ei ole yksityiskohtaisesti suunniteltu, mutta niiden yleinen teoreettinen KontrolliPalikka-luokka kuvaa teoriassa kaikkia mahdollisia control-luokkia.

3.3 View

View-luokat sijaitsevat `fi.helsinki.cs.ilpo.view` -pakkauksessa. Kyseessä on graafisen käyttöliittymän komponentit sisältävät luokat. View-luokat muodostuvat lopullisessa toteutuksessa kahdenlaisista luokista: valmiista Swing-komponenteista perittävistä luokista sekä puhtaasti grafiikalla tehtävistä luokista. Varsinkaan kaikkia Swing-ratkaisuun liittyviä luokkia ei ole vielä tarkasti suunniteltu.

Kuva 2: Luokkarakenne eli järjestelmän ydin



3.4 IO ja DB

Tiedostoja käsittelevät luokat ovat omassa pakkauksessaan (fi.helsinki.cs.ilpo.io). Pakkauksessa on luokat tulostiedostojen (opiskelijanumerolista ja valmis ryhmittely) kirjoitusta varten sekä työvälaineet, joilla keskeneräisiä ryhmäjakoja voidaan tallettaa ja ladata. Tietokantayhteys, kantaan tehtävät kyselyt sekä kyselyjen käsittely sijaitsevat fi.helsinki.cs.ilpo.db -pakkauksen luokissa. Paketteihin liittyvät luokat esitellään kuvassa 3.

4 Käyttöliittymä

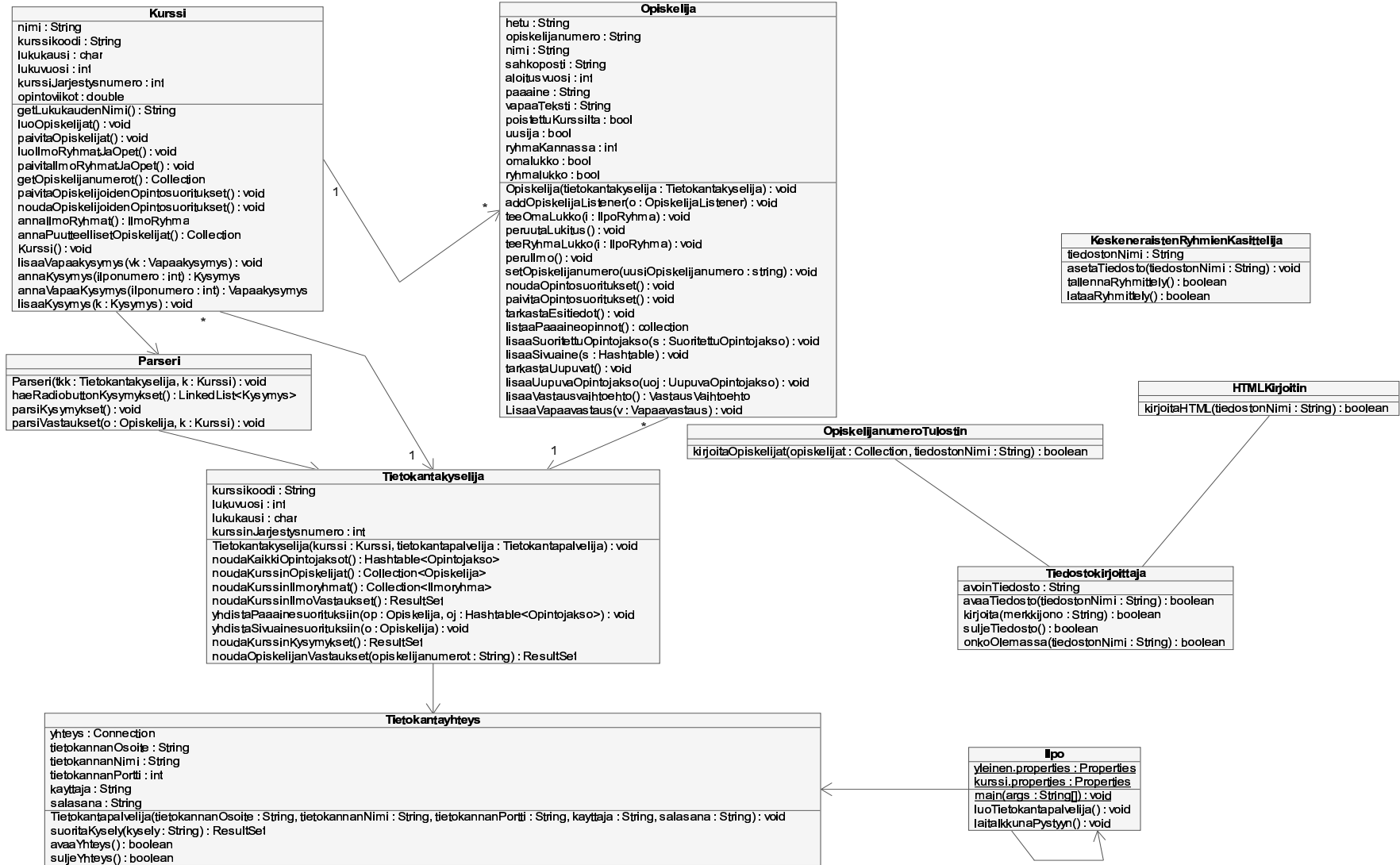
Ilpon käyttöliittymän suunnitelma löytyy määrittelydokumentin liitteestä. Tähän ei oteta tarkemmin kantaa tässä dokumentissa. Tässä dokumentissa käsitellään vain käyttöliittymän toteutusta.

Käyttöliittymän fyysinen rakenne koostuu kolmesta osasta: vasemmalla ylhäällä olevasta sort-solusta, vasemmalla alhaalla olevasta detail-solusta ja oikealla olevasta ryhmäsolusta. Sort-solu koostuu kahdesta tab-lehdestä: opiskelijoiden sort-solusta ja henkilökunnan sort-solusta. Opiskelijoiden sort-solu sisältää kurssille ilmoittautuneita opiskelijoita esittäviä palloja, ja tämä toteutetaan kokonaan grafiikalla. Henkilökunnan sort-solu sisältää listan kurssille ilmoitetuista henkilökuntalaisista, ja tämä toteutetaan swing-komponenteilla. Detail-solussa on joko tarkkaa tietoa yhdestä valitusta opiskelijasta tai vähemmän tarkkaa tietoa useasta valitusta opiskelijasta. Nämä toteutetaan swing-komponenteilla. Ryhmäsolu toteutetaan kokonaan grafiikalla. Se sisältää ryhmiä sekä niissä olevia opiskelijoita kuvaavia palloja ja henkilökuntalaisia kuvaavia komponentteja.

Varsinkaan grafiikalla toteutettavia osia ei ole tarkasti suunniteltu, koska kenelläkään ryhmäläisellä ei ole vankkaa kokemusta tai tietoa, kuinka se pitäisi tarkalleen ottaen tehdä. Swing-komponenteilla toteutettavien osien suunnitelmat taas ovat vajaita, koska swingin rakenne pakottaa käyttämään todella monia sisäkkäisiä luokkia (kuten JScrollPane, joka mahdollistaa näytön skrollauksen). Näitä ylimääräisiä luokkia ei ole suunnitelmiin mitenkään sisällytetty. Käyttöliittymäsuunnittelun pohjalla on ryhmämme oma tulkina model-view-control (MVC) -ratkaisusta. MVC on suunnittelumalli, jonka idea on jakaa järjestelmän toiminnallisuus kolmeen osaan: model-osassa on järjestelmän looginen toiminnallisuus (sekä tietosisältö), view-osassa on käyttäjälle esitettävä osa (eli graafisen käyttöliittymän komponentit, kuten napit ja muut napiskuukkelit) ja control-osassa on käyttäjän syötteiden prosessointi.

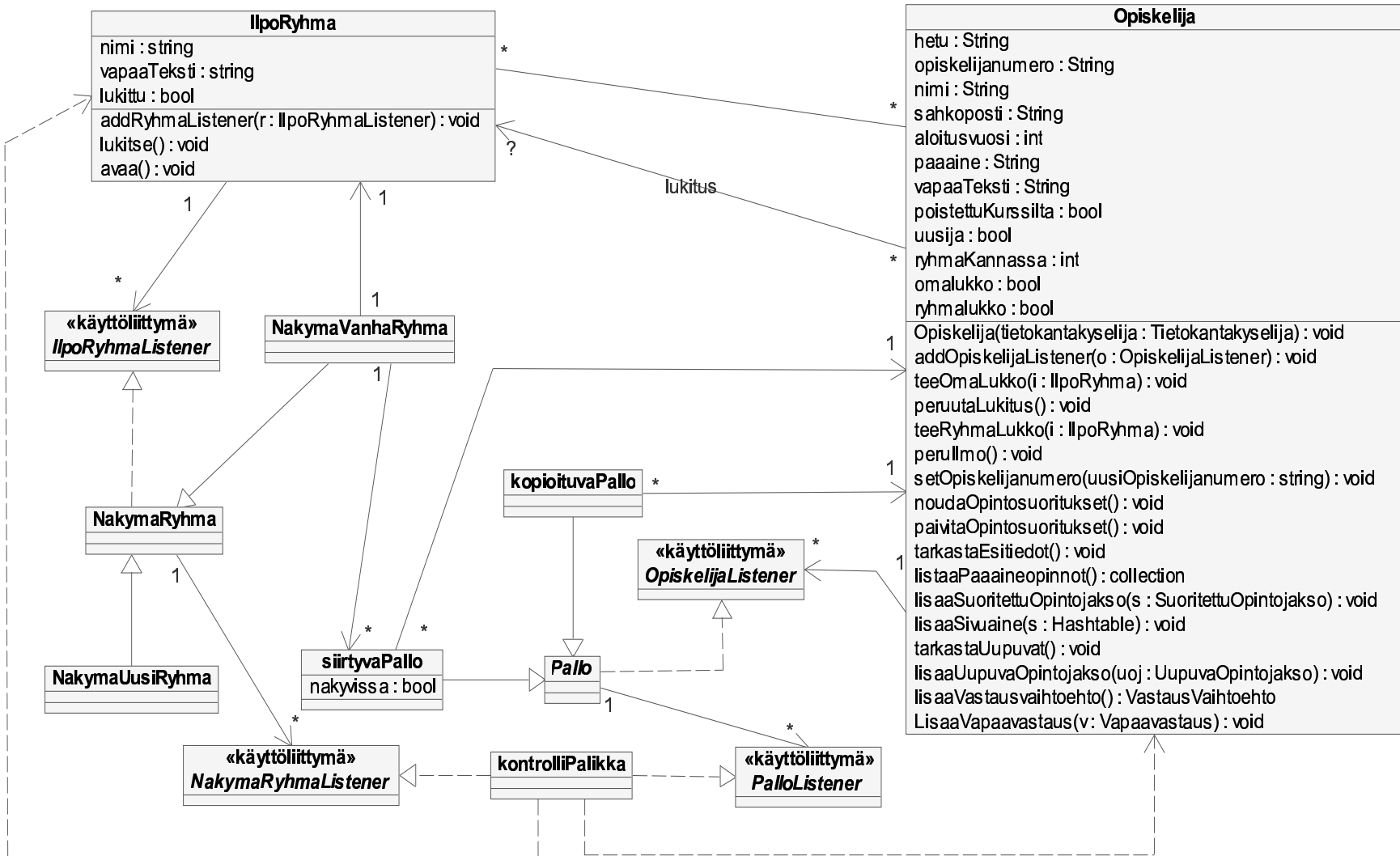
Ilpo on jaettu pakkauksiin siten, että fi.helsinki.cs.ilpo -pakkauksessa on model-luokat, fi.helsinki.cs.ilpo.view -pakkauksessa on view-luokat ja fi.helsinki.cs.ilpo.control -pakkauksessa on control-luokat. Käyttöliittymän luokkien välisiä yhteyksiä tarkastellaan kuvissa 4, 5 sekä 6. Kaikki model-luokat on kuvattu täydellisesti. View-luokat on kuvattu niin tarkasti, kuin suunnittelun kannalta on ollut järkevää, eli jokaista Swing-komponenttien perintää ja muita suoraan Swingiin liittyviä ominaisuuksia ei ole dokumentoitu. Yhtäkään yksittäistä control-luokkaa ei ole käytännössä lainkaan suunniteltu, mutta yleisesti kaikkien control-luokkien toiminta on sen sijaan suunniteltu. KontrolliPalikka -luokka kuvaa

Kuva 3: Tietokantaan ja tiedostoihin liittyvät luokat

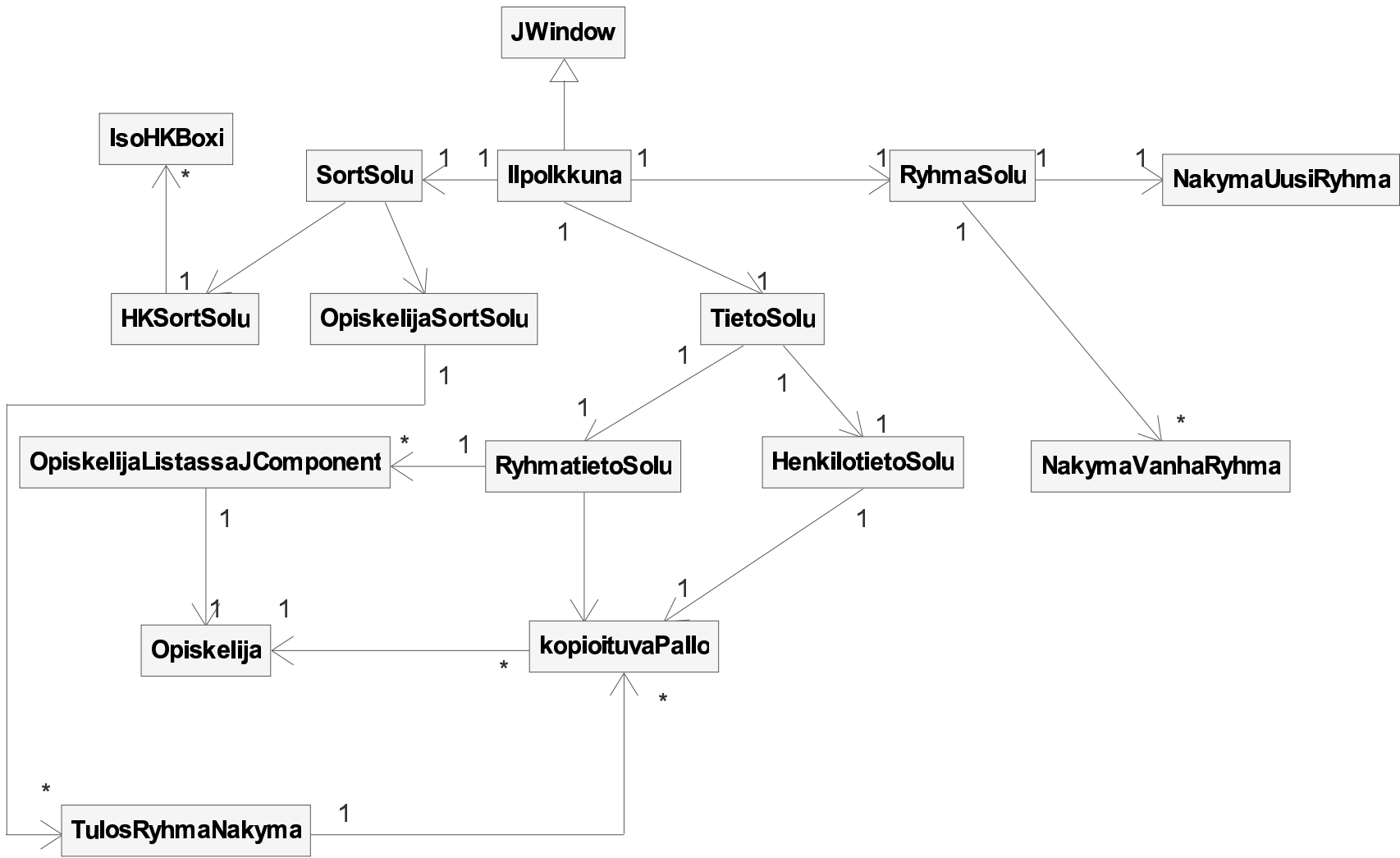


kaikkia teoriassa mahdollisia kontrolli-luokkia.

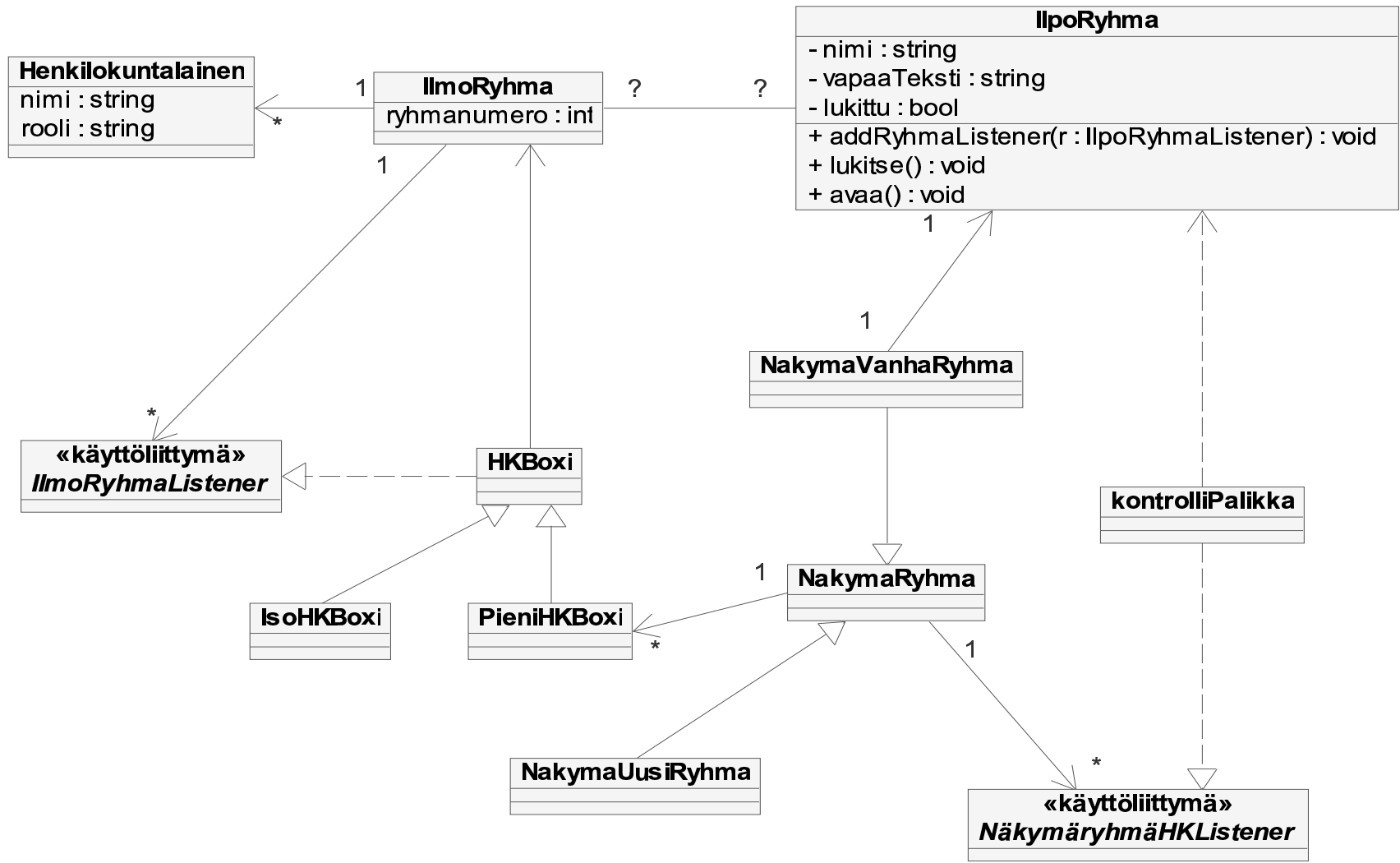
Kuva 4: Käyttöliittymän sisäiset rakenteet



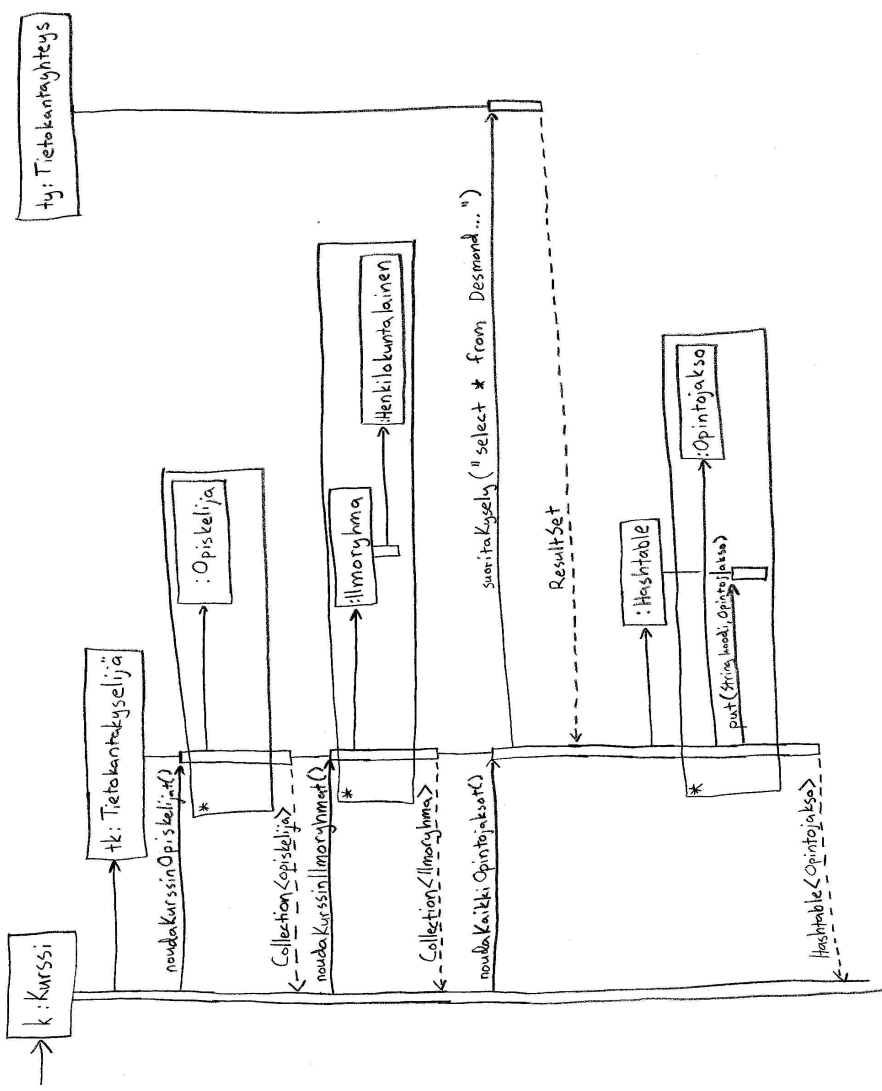
Kuva 5: Käyttöliittymän graafisiin komponentteihin liittyvät luokat



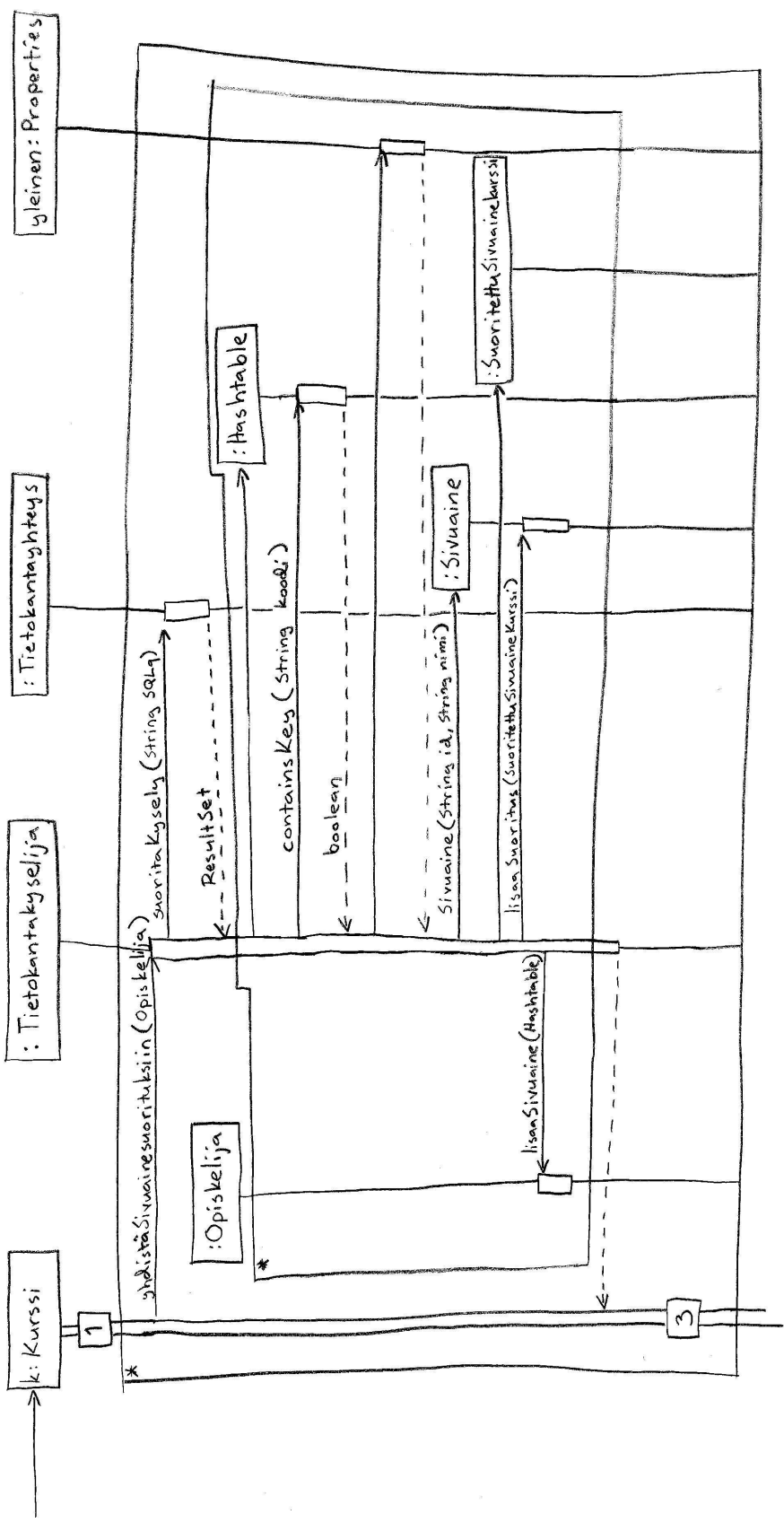
Kuva 6: Henkilökunnan kiinnittämiseen ilporyhmiin tarvittavat luokat



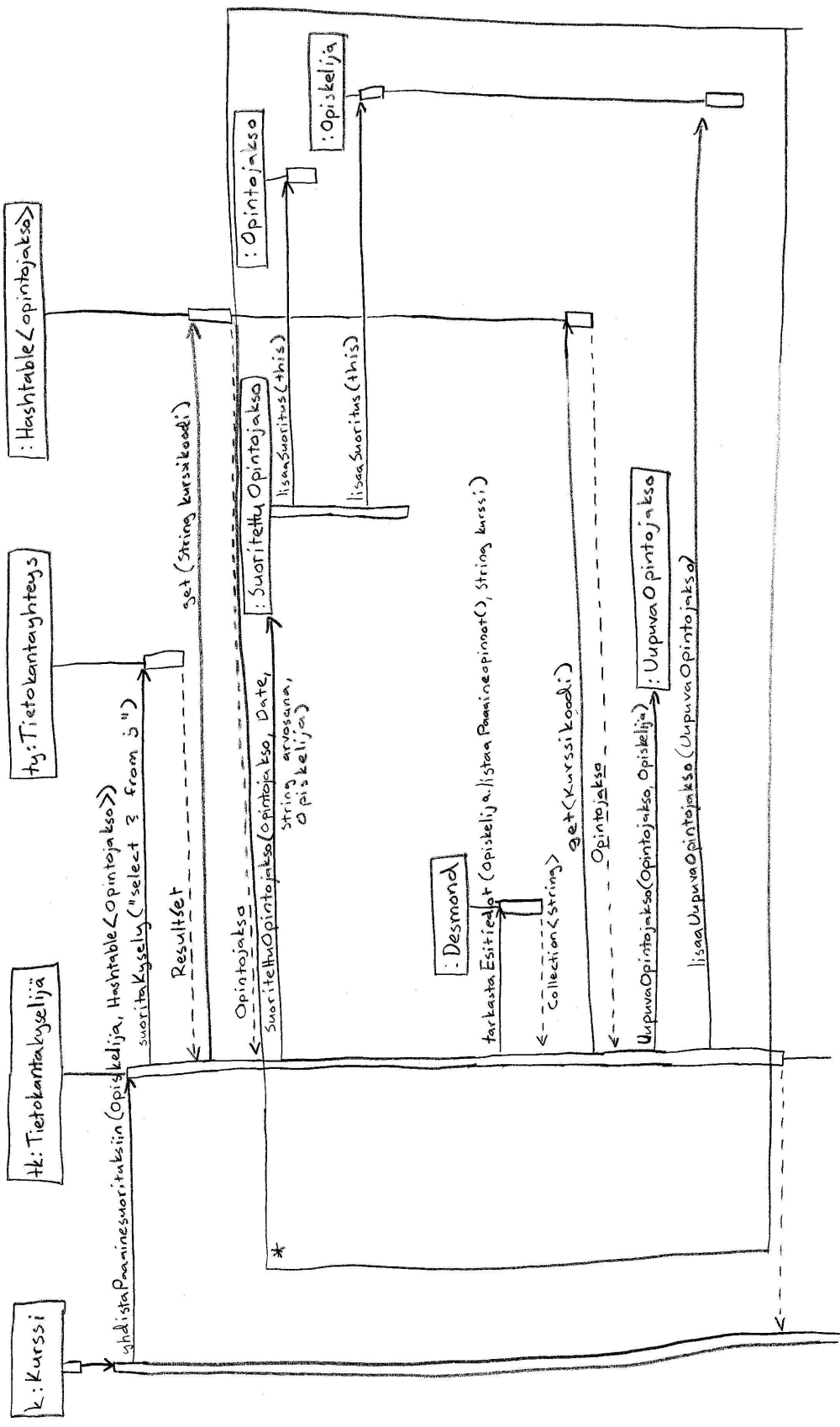
Liite 1. Sekvenssikaaviot

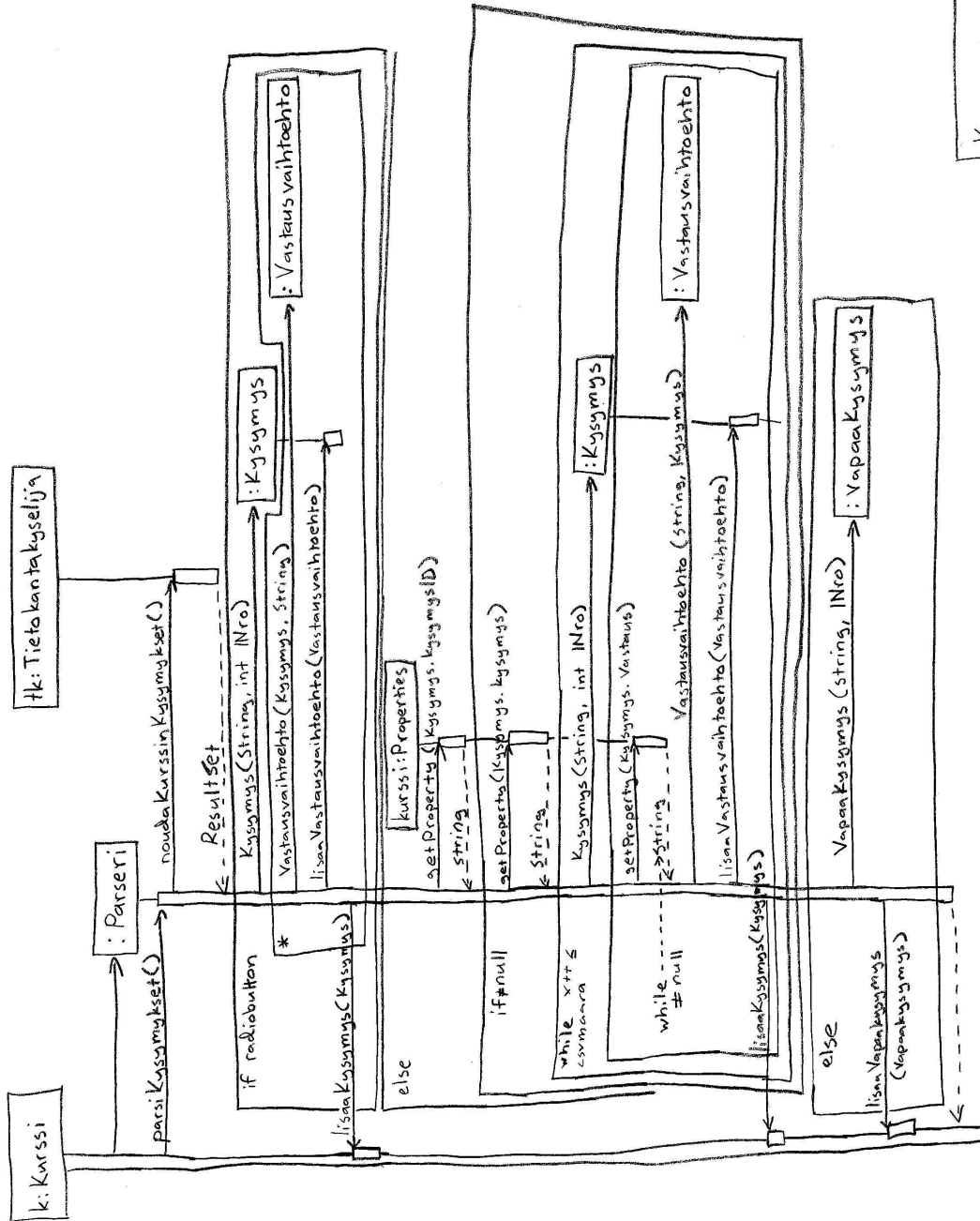


Opiskelijoiden, ilmoryhmien ja opintojaksoiden nouto

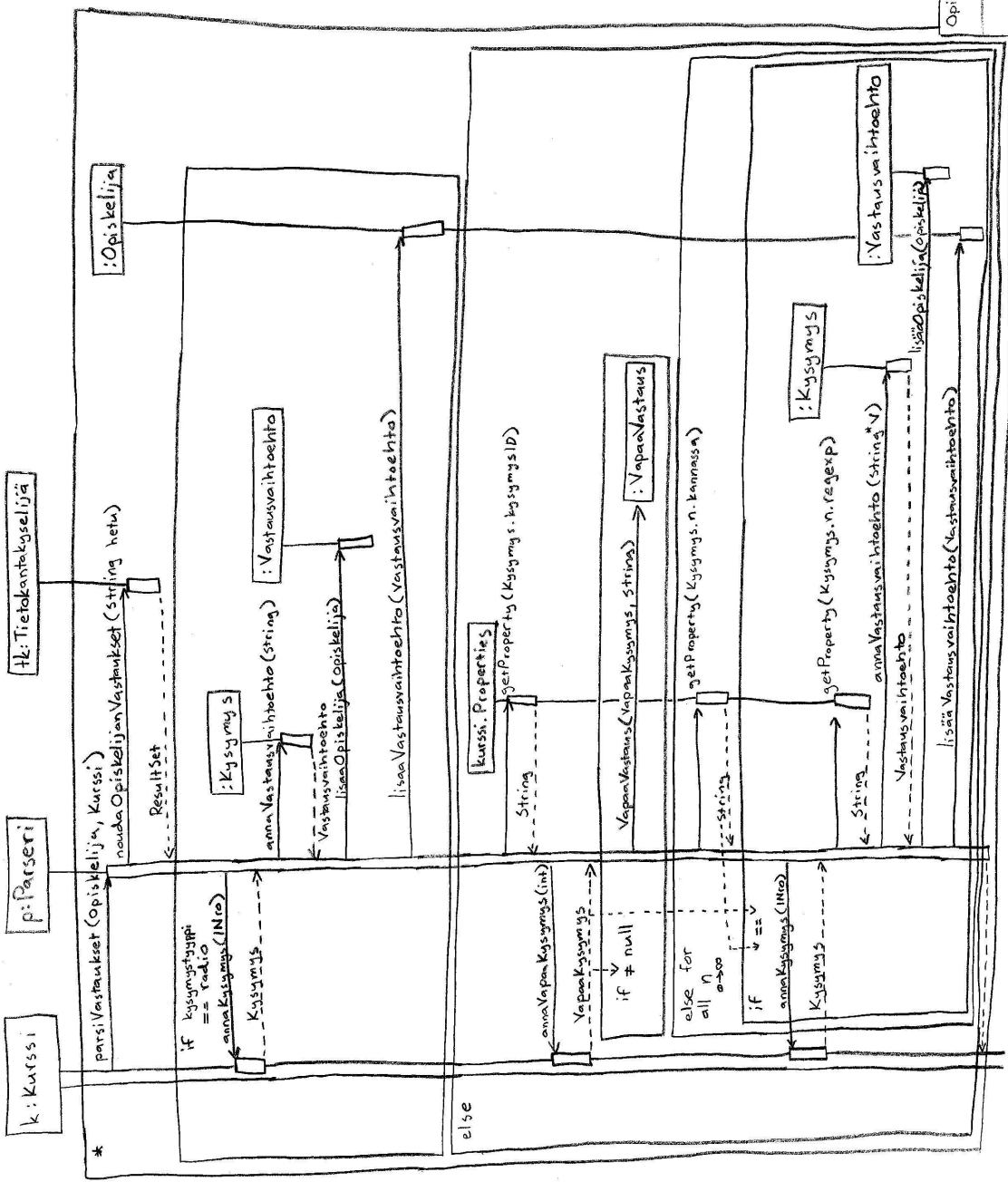


Sivunresurssien yhdistäminen 2





Kysymysten Haku



* Parsitaan propertiesista vastauksen numero vastauksiksi Stringiksi.

Opiskelijan ilmoitusten haku

Liite 2. Kurssi.properties-tiedosto

Esimerkki kurssi.properties-tiedostosta. Luettavuuden parantamiseksi osa asetuksista on jaettu usealle riville, todellisuudessa näin ei toimita.

```
koodi = 58110-3
kurssiJarjestysnumero = 1
lukukausi = kevat
lukuvuosi = 2005

kysymys.1.kannassa = 1
kysymys.2.kannassa = 3
kysymys.3.kannassa = 6
kysymys.3.preferenssi = 1
kysymys.3.tyyppi = csv
kysymys.3.vv.1 = algoritmit, tietorakenteet
kysymys.3.vv.2 = laskennan teoria
kysymys.3.vv.3 = hahmontunnistus
kysymys.3.vv.4 = tietokonegrafiikka
kysymys.3.vv.5 = tekoäly
kysymys.3.vv.6 = koneoppiminen
kysymys.3.vv.7 = data-analyysin laskennalliset menetelmät
kysymys.3.vv.8 = ohjelmointi, ohjelmointikielet, olio-ohjelmointi
kysymys.3.vv.9 = ohjelmiston rakenne (arkkitehtuuri)
kysymys.3.vv.10 = ohjelmiston laatu, ohjelmiston mittaaminen
kysymys.3.vv.11 = ohjelmiston kehitysprosessi, ohjelmistotuotanto
kysymys.3.vv.12 = testaus
kysymys.3.vv.13 = tietoliikenne
kysymys.3.vv.14 = mobiilijärjestelmät
kysymys.3.vv.15 = käyttöjärjestelmät
kysymys.3.vv.16 = tietokoneen rakenne
kysymys.3.vv.17 = hajautetut sovellusalustat ja sovellukset
kysymys.3.vv.18 = sähköisen liiketoiminnan tukijärjestelmät
kysymys.3.vv.19 = älykkäästi sisältöä käsittelevät web-palvelut
kysymys.3.vv.20 = dokumenttien käsittelymenetelmät (mm. XML)
kysymys.3.vv.21 = tiedonhakumenetelmät
kysymys.3.vv.22 = tiedon louhinta
kysymys.3.vv.23 = bioinformatiikka
kysymys.3.vv.24 = kieliteknologia
kysymys.3.vv.25 = tietokannat
kysymys.3.vv.26 = käyttöliittymät
kysymys.3.vv.27 = opettajalinja, aihepiiristä toivotaan
tarkennusta lisätiedoissa
kysymys.3.vv.28 = muu, aihepiiri tarkennettava lisätiedoissa
(esimerkiksi sovelletun tietojenkäsittelyn osalta sovellusala)

kysymys.4.kannassa = 6
kysymys.4.preferenssi = 2
```


kysymys.4.tyyppi = csv
kysymys.4.vv.1 = algoritmit, tietorakenteet
kysymys.4.vv.2 = laskennan teoria
kysymys.4.vv.3 = hahmontunnistus
kysymys.4.vv.4 = tietokonegrafiikka
kysymys.4.vv.5 = tekoäly
kysymys.4.vv.6 = koneoppiminen
kysymys.4.vv.7 = data-analyysin laskennalliset menetelmät
kysymys.4.vv.8 = ohjelmointi, ohjelmointikielet, olio-ohjelmointi
kysymys.4.vv.9 = ohjelmiston rakenne (arkkitehtuuri)
kysymys.4.vv.10 = ohjelmiston laatu, ohjelmiston mittaaminen
kysymys.4.vv.11 = ohjelmiston kehitysprosessi, ohjelmistotuotanto
kysymys.4.vv.12 = testaus
kysymys.4.vv.13 = tietoliikenne
kysymys.4.vv.14 = mobiilijärjestelmät
kysymys.4.vv.15 = käyttöjärjestelmät
kysymys.4.vv.16 = tietokoneen rakenne
kysymys.4.vv.17 = hajautetut sovellusalustat ja sovellukset
kysymys.4.vv.18 = sähköisen liiketoiminnan tukijärjestelmät
kysymys.4.vv.19 = älykkäästi sisältöä käsittelevät web-palvelut
kysymys.4.vv.20 = dokumenttien käsittelymenetelmät (mm. XML)
kysymys.4.vv.21 = tiedonhakumenetelmät
kysymys.4.vv.22 = tiedon louhinta
kysymys.4.vv.23 = bioinformatiikka
kysymys.4.vv.24 = kieliteknologia
kysymys.4.vv.25 = tietokannat
kysymys.4.vv.26 = käyttöliittymät
kysymys.4.vv.27 = opettajalinja, aihepiiristä toivotaan
tarkennusta lisätiedoissa
kysymys.4.vv.28 = muu, aihepiiri tarkennettava lisätiedoissa
(esimerkiksi sovelletun tietojenkäsittelyn osalta sovellusala)

kysymys.5.kannassa = 6
kysymys.5.preferenssi = 3
kysymys.5.tyyppi = csv
kysymys.5.vv.1 = algoritmit, tietorakenteet
kysymys.5.vv.2 = laskennan teoria
kysymys.5.vv.3 = hahmontunnistus
kysymys.5.vv.4 = tietokonegrafiikka
kysymys.5.vv.5 = tekoäly
kysymys.5.vv.6 = koneoppiminen
kysymys.5.vv.7 = data-analyysin laskennalliset menetelmät
kysymys.5.vv.8 = ohjelmointi, ohjelmointikielet, olio-ohjelmointi
kysymys.5.vv.9 = ohjelmiston rakenne (arkkitehtuuri)
kysymys.5.vv.10 = ohjelmiston laatu, ohjelmiston mittaaminen
kysymys.5.vv.11 = ohjelmiston kehitysprosessi, ohjelmistotuotanto
kysymys.5.vv.12 = testaus

kysymys.5.vv.13 = tietoliikenne
kysymys.5.vv.14 = mobiilijärjestelmät
kysymys.5.vv.15 = käyttöjärjestelmät
kysymys.5.vv.16 = tietokoneen rakenne
kysymys.5.vv.17 = hajautetut sovellusalustat ja sovellukset
kysymys.5.vv.18 = sähköisen liiketoiminnan tukijärjestelmät
kysymys.5.vv.19 = älykkäästi sisältöä käsittelevät web-palvelut
kysymys.5.vv.20 = dokumenttien käsittelymenetelmät (mm. XML)
kysymys.5.vv.21 = tiedonhakumenetelmät
kysymys.5.vv.22 = tiedon louhinta
kysymys.5.vv.23 = bioinformatiikka
kysymys.5.vv.24 = kieliteknologia
kysymys.5.vv.25 = tietokannat
kysymys.5.vv.26 = käyttöliittymät
kysymys.5.vv.27 = opettajalinja, aihepiiristä toivotaan
tarkennusta lisätiedoissa
kysymys.5.vv.28 = muu, aihepiiri tarkennettava lisätiedoissa
(esimerkiksi sovelletun tietojenkäsittelyn osalta sovellusala)

kysymys.6.kannassa = 6
kysymys.6.preferenssi = 4
kysymys.6.tyyppi = csv
kysymys.6.vv.1 = algoritmit, tietorakenteet
kysymys.6.vv.2 = laskennan teoria
kysymys.6.vv.3 = hahmontunnistus
kysymys.6.vv.4 = tietokonegrafiikka
kysymys.6.vv.5 = tekoäly
kysymys.6.vv.6 = koneoppiminen
kysymys.6.vv.7 = data-analyysin laskennalliset menetelmät
kysymys.6.vv.8 = ohjelmointi, ohjelmointikielet, olio-ohjelmointi
kysymys.6.vv.9 = ohjelmiston rakenne (arkkitehtuuri)
kysymys.6.vv.10 = ohjelmiston laatu, ohjelmiston mittaaminen
kysymys.6.vv.11 = ohjelmiston kehitysprosessi, ohjelmistotuotanto
kysymys.6.vv.12 = testaus
kysymys.6.vv.13 = tietoliikenne
kysymys.6.vv.14 = mobiilijärjestelmät
kysymys.6.vv.15 = käyttöjärjestelmät
kysymys.6.vv.16 = tietokoneen rakenne
kysymys.6.vv.17 = hajautetut sovellusalustat ja sovellukset
kysymys.6.vv.18 = sähköisen liiketoiminnan tukijärjestelmät
kysymys.6.vv.19 = älykkäästi sisältöä käsittelevät web-palvelut
kysymys.6.vv.20 = dokumenttien käsittelymenetelmät (mm. XML)
kysymys.6.vv.21 = tiedonhakumenetelmät
kysymys.6.vv.22 = tiedon louhinta
kysymys.6.vv.23 = bioinformatiikka
kysymys.6.vv.24 = kieliteknologia
kysymys.6.vv.25 = tietokannat

kysymys.6.vv.26 = käyttöliittymät
kysymys.6.vv.27 = opettajalinja, aihepiiristä toivotaan
tarkennusta lisätiedoissa
kysymys.6.vv.28 = muu, aihepiiri tarkennettava lisätiedoissa
(esimerkiksi sovelletun tietojenkäsittelyn osalta sovellusala)

kysymys.7.kannassa = 6
kysymys.7.preferenssi = 5
kysymys.7.tyyppi = csv
kysymys.7.vv.1 = algoritmit, tietorakenteet
kysymys.7.vv.2 = laskennan teoria
kysymys.7.vv.3 = hahmontunnistus
kysymys.7.vv.4 = tietokonegrafiikka
kysymys.7.vv.5 = tekoäly
kysymys.7.vv.6 = koneoppiminen
kysymys.7.vv.7 = data-analyysin laskennalliset menetelmät
kysymys.7.vv.8 = ohjelmointi, ohjelmointikielet, olio-ohjelmointi
kysymys.7.vv.9 = ohjelmiston rakenne (arkkitehtuuri)
kysymys.7.vv.10 = ohjelmiston laatu, ohjelmiston mittaaminen
kysymys.7.vv.11 = ohjelmiston kehitysprosessi, ohjelmistotuotanto
kysymys.7.vv.12 = testaus
kysymys.7.vv.13 = tietoliikenne
kysymys.7.vv.14 = mobiilijärjestelmät
kysymys.7.vv.15 = käyttöjärjestelmät
kysymys.7.vv.16 = tietokoneen rakenne
kysymys.7.vv.17 = hajautetut sovellusalustat ja sovellukset
kysymys.7.vv.18 = sähköisen liiketoiminnan tukijärjestelmät
kysymys.7.vv.19 = älykkäästi sisältöä käsittelevät web-palvelut
kysymys.7.vv.20 = dokumenttien käsittelymenetelmät (mm. XML)
kysymys.7.vv.21 = tiedonhakumenetelmät
kysymys.7.vv.22 = tiedon louhinta
kysymys.7.vv.23 = bioinformatiikka
kysymys.7.vv.24 = kieliteknologia
kysymys.7.vv.25 = tietokannat
kysymys.7.vv.26 = käyttöliittymät
kysymys.7.vv.27 = opettajalinja, aihepiiristä toivotaan
tarkennusta lisätiedoissa
kysymys.7.vv.28 = muu, aihepiiri tarkennettava lisätiedoissa
(esimerkiksi sovelletun tietojenkäsittelyn osalta sovellusala)

kysymys.8.kannassa = 2
kysymys.9.kannassa = 7
kysymys.10.kannassa = 15

html.otsikko = Kevään 2005 ohjelmistotuotantoprojektien
alustava ryhmäjako
html.ingressi = Seuraavassa on 12.01.2005 päivätty lista

kevään 2005 ohjelmistotuotantoprojekteista.
html.kokoontumisajat = Ryhmät aloittavat kokoontumisensa
viikolla 4 (24.1.2005 - 28.1.2005).
Projektit päättyvät viimeistään
viikolla 18 (2.5.2005 - 6.5.2005).
html.jonotusotsikko = Jonotuslista
html.jonotusohjeet = Jonotuslistalle on sijoitettu
opiskelijat, joilla on ryhmäjako
laadittaessa ollut puutteita kurssin
esitietovaatimusten suhteen.
Opiskelijoita voidaan sijoittaa
listalta ryhmiin, mikäli he saavat
puuttuvat suoritukset valmiiksi ennen
ryhmien toiminnan alkamista. Jos löydät
nimesi listalta ja mielestäsi
esitietovaatimukset ovat kohdaltasi
kunnossa, ota mahdollisimman pian
yhteyttä osoitteeseen
ohtu@cs.helsinki.fi
html.spostiosoite = ohtu@cs.helsinki.fi

Liite 3. Yleinen.properties-tiedosto

Esimerkki yleinen.properties-tiedostosta

```
tk.osoite = kontti.helsinki.fi  
tk.portti = 1522  
tk.nimi = tkta
```

Contents

1	Pakkaus fi.helsinki.cs.ilpo	2
1.1	Luokka Henkilokuntalainen	3
1.1.1	Kuvaus	3
1.1.2	Konstruktorien yhteenveto	3
1.1.3	Metodien yhteenveto	3
1.1.4	Konstruktorit	3
1.1.5	Metodit	3
1.2	Luokka IlmoRyhma	4
1.2.1	Kuvaus	4
1.2.2	Konstruktorien yhteenveto	4
1.2.3	Metodien yhteenveto	4
1.2.4	Konstruktorit	4
1.2.5	Metodit	4
1.3	Luokka Ilpo	5
1.3.1	Kuvaus	5
1.3.2	Metodien yhteenveto	5
1.3.3	Metodit	5
1.4	Luokka IlpoRyhma	6
1.4.1	Kuvaus	6
1.4.2	Konstruktorien yhteenveto	6
1.4.3	Metodien yhteenveto	7
1.4.4	Konstruktorit	7
1.4.5	Metodit	7
1.5	Luokka Kurssi	8
1.5.1	Kuvaus	8
1.5.2	Konstruktorien yhteenveto	8
1.5.3	Metodien yhteenveto	8
1.5.4	Konstruktorit	9
1.5.5	Metodit	10
1.6	Luokka Kysymys	13
1.6.1	Kuvaus	14
1.6.2	Konstruktorien yhteenveto	14
1.6.3	Metodien yhteenveto	14

1.6.4	Konstruktorit	14
1.6.5	Metodit	14
1.7	Luokka Opintojakso	15
1.7.1	Kuvaus	15
1.7.2	Konstruktorien yhteenveto	15
1.7.3	Metodien yhteenveto	15
1.7.4	Konstruktorit	16
1.7.5	Metodit	16
1.8	Luokka Opiskelija	16
1.8.1	Kuvaus	17
1.8.2	Konstruktorien yhteenveto	17
1.8.3	Metodien yhteenveto	17
1.8.4	Konstruktorit	18
1.8.5	Metodit	18
1.9	Luokka Sivuaine	22
1.9.1	Kuvaus	22
1.9.2	Konstruktorien yhteenveto	23
1.9.3	Metodien yhteenveto	23
1.9.4	Konstruktorit	23
1.9.5	Metodit	23
1.10	Luokka SuoritettuOpintojakso	24
1.10.1	Kuvaus	24
1.10.2	Konstruktorien yhteenveto	24
1.10.3	Metodien yhteenveto	24
1.10.4	Konstruktorit	24
1.10.5	Metodit	25
1.11	Luokka SuoritettuSivuainekurssi	26
1.11.1	Kuvaus	26
1.11.2	Konstruktorien yhteenveto	26
1.11.3	Konstruktorit	26
1.12	Luokka UupuvaOpintojakso	26
1.12.1	Kuvaus	26
1.12.2	Konstruktorien yhteenveto	27
1.12.3	Metodien yhteenveto	27
1.12.4	Konstruktorit	27
1.12.5	Metodit	27
1.13	Luokka Vapaakysymys	28
1.13.1	Kuvaus	28
1.13.2	Konstruktorien yhteenveto	28
1.13.3	Metodien yhteenveto	28
1.13.4	Konstruktorit	28
1.13.5	Metodit	28
1.14	Luokka Vapaavastaus	29
1.14.1	Kuvaus	29

1.14.2	Konstruktorien yhteenveto	29
1.14.3	Konstruktorit	29
1.15	Luokka Vastausvaihtoehto	29
1.15.1	Kuvaus	29
1.15.2	Konstruktorien yhteenveto	29
1.15.3	Metodien yhteenveto	30
1.15.4	Konstruktorit	30
1.15.5	Metodit	30
2	Pakkaus fi.helsinki.cs.ilpo.control	31
2.1	Luokka IlmoRyhmaListener	31
2.1.1	Kuvaus	31
2.1.2	Kaikki tunnetut alirajapinnat	32
2.1.3	Kaikki luokat, joiden tiedetään toteuttavan rajapinnan . . .	32
2.2	Luokka IlpoRyhmaListener	32
2.2.1	Kuvaus	32
2.2.2	Kaikki tunnetut alirajapinnat	32
2.2.3	Kaikki luokat, joiden tiedetään toteuttavan rajapinnan . . .	32
2.3	Luokka NakymaRyhmaHKListener	32
2.3.1	Kuvaus	32
2.3.2	Kaikki tunnetut alirajapinnat	32
2.3.3	Kaikki luokat, joiden tiedetään toteuttavan rajapinnan . . .	32
2.4	Luokka NakymaRyhmaListener	32
2.4.1	Kuvaus	33
2.4.2	Kaikki tunnetut alirajapinnat	33
2.4.3	Kaikki luokat, joiden tiedetään toteuttavan rajapinnan . . .	33
2.5	Luokka OpiskelijaListener	33
2.5.1	Kuvaus	33
2.5.2	Kaikki tunnetut alirajapinnat	33
2.5.3	Kaikki luokat, joiden tiedetään toteuttavan rajapinnan . . .	33
2.6	Luokka PalloListener	33
2.6.1	Kuvaus	33
2.6.2	Kaikki tunnetut alirajapinnat	33
2.6.3	Kaikki luokat, joiden tiedetään toteuttavan rajapinnan . . .	33
2.7	Luokka KontrolliPalikka	34
2.7.1	Kuvaus	34
2.7.2	Konstruktorien yhteenveto	34
2.7.3	Konstruktorit	34
2.8	Luokka KontrolliYlenkatsoUupuva	34
2.8.1	Kuvaus	34
2.8.2	Konstruktorien yhteenveto	34
2.8.3	Metodien yhteenveto	35
2.8.4	Konstruktorit	35
2.8.5	Metodit	35

3	Pakkaus fi.helsinki.cs.ilpo.view	36
3.1	Luokka AutentikointiIkkuna	37
3.1.1	Kuvaus	37
3.1.2	Kenttien yhteenveto	38
3.1.3	Konstruktorien yhteenveto	38
3.1.4	Metodien yhteenveto	38
3.1.5	Kentät	38
3.1.6	Konstruktorit	38
3.1.7	Metodit	38
3.2	Luokka HenkilotietoSolu	38
3.2.1	Kuvaus	39
3.2.2	Konstruktorien yhteenveto	39
3.2.3	Konstruktorit	39
3.3	Luokka HKBoxi	39
3.3.1	Kuvaus	39
3.3.2	Kaikki tunnetut aliluokat	39
3.3.3	Konstruktorien yhteenveto	39
3.3.4	Konstruktorit	39
3.4	Luokka HKSortSolu	40
3.4.1	Kuvaus	40
3.4.2	Konstruktorien yhteenveto	40
3.4.3	Konstruktorit	40
3.5	Luokka IsoHKBoxi	40
3.5.1	Kuvaus	40
3.5.2	Konstruktorien yhteenveto	40
3.5.3	Konstruktorit	40
3.6	Luokka KopioituvaPallo	41
3.6.1	Kuvaus	41
3.6.2	Konstruktorien yhteenveto	41
3.6.3	Konstruktorit	41
3.7	Luokka NakymaRyhma	41
3.7.1	Kuvaus	41
3.7.2	Kaikki tunnetut aliluokat	42
3.7.3	Konstruktorien yhteenveto	42
3.7.4	Konstruktorit	42
3.8	Luokka NakymaUusiRyhma	42
3.8.1	Kuvaus	42
3.8.2	Konstruktorien yhteenveto	42
3.8.3	Konstruktorit	42
3.9	Luokka NakymaVanhaRyhma	42
3.9.1	Kuvaus	43
3.9.2	Konstruktorien yhteenveto	43
3.9.3	Konstruktorit	43
3.10	Luokka OpiskelijaListassaJComponent	43

3.10.1	Kuvaus	43
3.10.2	Konstruktorien yhteenveto	43
3.10.3	Konstruktorit	43
3.11	Luokka OpiskelijaSortSolu	44
3.11.1	Kuvaus	44
3.11.2	Konstruktorien yhteenveto	44
3.11.3	Metodien yhteenveto	44
3.11.4	Konstruktorit	44
3.11.5	Metodit	44
3.12	Luokka Pallo	45
3.12.1	Kuvaus	45
3.12.2	Kaikki tunnetut aliluokat	45
3.12.3	Konstruktorien yhteenveto	45
3.12.4	Konstruktorit	45
3.13	Luokka PieniHKBoxi	45
3.13.1	Kuvaus	45
3.13.2	Konstruktorien yhteenveto	45
3.13.3	Konstruktorit	46
3.14	Luokka PoistaOpiskelijaButton	46
3.14.1	Kuvaus	46
3.14.2	Konstruktorien yhteenveto	46
3.14.3	Konstruktorit	46
3.15	Luokka PoistaRyhmaButton	46
3.15.1	Kuvaus	46
3.15.2	Konstruktorien yhteenveto	46
3.15.3	Konstruktorit	47
3.16	Luokka PoistaRyhmasterButton	47
3.16.1	Kuvaus	47
3.16.2	Konstruktorien yhteenveto	47
3.16.3	Konstruktorit	47
3.17	Luokka RyhmaSolu	47
3.17.1	Kuvaus	47
3.17.2	Konstruktorien yhteenveto	47
3.17.3	Metodien yhteenveto	47
3.17.4	Konstruktorit	48
3.17.5	Metodit	48
3.18	Luokka RyhmatietoSolu	48
3.18.1	Kuvaus	48
3.18.2	Konstruktorien yhteenveto	48
3.18.3	Konstruktorit	48
3.19	Luokka SiirtyvaPallo	48
3.19.1	Kuvaus	49
3.19.2	Konstruktorien yhteenveto	49
3.19.3	Konstruktorit	49

3.20	Luokka SortSolu	49
3.20.1	Kuvaus	49
3.20.2	Konstruktorien yhteenveto	49
3.20.3	Konstruktorit	49
3.21	Luokka TietoSolu	50
3.21.1	Kuvaus	50
3.21.2	Konstruktorien yhteenveto	50
3.21.3	Konstruktorit	50
3.22	Luokka TulosRyhmaNakyma	50
3.22.1	Kuvaus	50
3.22.2	Konstruktorien yhteenveto	50
3.22.3	Konstruktorit	50
3.23	Luokka YlenkatsoJButton	51
3.23.1	Kuvaus	51
3.23.2	Konstruktorien yhteenveto	51
3.23.3	Konstruktorit	51
4	Pakkaus fi.helsinki.cs.ilpo.db	52
4.1	Luokka Parseri	52
4.1.1	Kuvaus	52
4.1.2	Konstruktorien yhteenveto	52
4.1.3	Metodien yhteenveto	52
4.1.4	Konstruktorit	53
4.1.5	Metodit	53
4.2	Luokka Tietokantakyselija	53
4.2.1	Kuvaus	54
4.2.2	Konstruktorien yhteenveto	54
4.2.3	Metodien yhteenveto	54
4.2.4	Konstruktorit	55
4.2.5	Metodit	55
4.3	Luokka Tietokantayhteys	58
4.3.1	Kuvaus	58
4.3.2	Konstruktorien yhteenveto	58
4.3.3	Metodien yhteenveto	58
4.3.4	Konstruktorit	58
4.3.5	Metodit	59
5	Pakkaus fi.helsinki.cs.ilpo.io	60
5.1	Luokka HTMLKirjoitin	60
5.1.1	Kuvaus	60
5.1.2	Metodien yhteenveto	60
5.1.3	Metodit	61
5.2	Luokka KeskeneraistenRyhmiemKasittelija	61
5.2.1	Kuvaus	61

5.2.2	Konstruktorien yhteenveto	61
5.2.3	Metodien yhteenveto	61
5.2.4	Konstruktorit	62
5.2.5	Metodit	62
5.3	Luokka OpiskelijanumeroTulostin	63
5.3.1	Kuvaus	63
5.3.2	Metodien yhteenveto	63
5.3.3	Metodit	63
5.4	Luokka Tiedostokirjoittaja	64
5.4.1	Kuvaus	64
5.4.2	Konstruktorien yhteenveto	64
5.4.3	Metodien yhteenveto	64
5.4.4	Konstruktorit	64
5.4.5	Metodit	64

Chapter 1

Pakkaus fi.helsinki.cs.ilpo

<i>Pakkaus Sisältö</i>	<i>Sivu</i>
Luokat	
Henkilökuntalainen	3
Henkilökunnan jäsen, joka liittyy ilpon tuottamiin lopullisiin ryhmiin.	
IlmoRyhma	4
IlmoRyhmat ovat peräisin Kurjesta.	
Ilpo	5
Class Ilpo	
IlpoRyhma	6
Ilpon sisäisen ryhmäjaon ryhmät.	
Kurssi	8
Järjestelmän kannalta keskeinen tietorakenne, johon liittyy kaikki yhden kurssin ryhmittelyn kannalta olennainen tieto.	
Kysymys	13
Ilmon monivalintakysymys ilpossa.	
Opintojakso	15
TKTL:n kurssi, jonka opiskelija on saattanut suorittaa.	
Opiskelija	16
Edustaa yhtä kurssille ilmoittautunutta opiskelijaa.	
Sivuaine	22
Opiskelijan yhden sivuaine-aineen suoritusten majapaikka.	
SuoritettuOpintojakso	24
Yhdistää opiskelijan ja tämän suorittaman tkl:n opintojakson toisiinsa.	
SuoritettuSivuainekurssi	26
Opiskelijan suorittama sivuainekurssi.	
UupuvaOpintojakso	26
Kurssin esitietoihin kuuluva opintojakso, jota opiskelija ei ole suorittanut.	
Vapaakysymys	28
Ilmoittatumisen yhteydessä kysytty kysymys, johon on voitu vastata vapaamuotoisesti.	

Vapaavastaus	29
Opiskelijan vastaus vapaakysymykseen.	
Vastausvaihtoehto	29
Vastausvaihtoehto sellaisen kysymyksen, johon on valmiit vastausvaihtoehdot.	

1.1 Luokka Henkilokuntalainen

Henkilökunnan jäsen, joka liittyy ilpon tuottamiin lopullisiin ryhmiin. Henkilöstä on tiedossa nimi ja rooli kurssin pienryhmässä. Luokka on tiukassa symbioosissa ilpoRyhma-luokan ilmentymien kanssa.

1.1.1 Kuvaus

```
public class Henkilokuntalainen
extends java.lang.Object
implements java.io.Serializable
```

1.1.2 Konstruktorien yhteenveto

Henkilokuntalainen(String, String) Luo uuden kurssiin liittyvän henkilökunnan jäsenen.

1.1.3 Metodien yhteenveto

toString() Palauttaa tekstiesityksen (nimi / rooli) henkilökunnan jäsenestä

1.1.4 Konstruktorit

- **Henkilokuntalainen**

```
public Henkilokuntalainen( java.lang.String nimi,
java.lang.String rooli )
```

- **Kuvaus**

Luo uuden kurssiin liittyvän henkilökunnan jäsenen.

- **Parametrit**

- * **nimi** – Kurssiin liittyvän henkilökunnan jäsenen nimi
- * **rooli** – Tehtävä, jota henkilökunnan jäsen hoitaa

1.1.5 Metodit

- **toString**

```
public java.lang.String toString( )
```

- **Kuvaus**
Palauttaa tekstiesityksen (nimi / rooli) henkilökunnan jäsenestä
- **Palauttaa** – Tekstiesitys

1.2 Luokka IlmoRyhma

IlmoRyhmat ovat peräisin Kurjesta. Niitä käytetään kun järjestelmän sisäiset ryhmät talletetaan tietokantaan.

1.2.1 Kuvaus

```
public class IlmoRyhma
extends java.lang.Object
implements java.io.Serializable
```

1.2.2 Konstruktorien yhteenveto

IlmoRyhma(int) Luo uuden numeron mukaisen Ilmoryhmän

1.2.3 Metodien yhteenveto

annaHenkilokuntalaiset() Palauttaa ryhmän työntekijät.
lisaaHenkilokuntalainen(Henkilokuntalainen) Lisää ryhmään työläisen.

1.2.4 Konstruktorit

- **IlmoRyhma**
`public IlmoRyhma(int ryhmanumero)`
 - **Kuvaus**
Luo uuden numeron mukaisen Ilmoryhmän
 - **Parametrit**
* `ryhmanumero` – Ryhmän numero Ilmossa.

1.2.5 Metodit

- **annaHenkilokuntalaiset**
`public java.util.Collection annaHenkilokuntalaiset()`
 - **Kuvaus**
Palauttaa ryhmän työntekijät.
 - **Palauttaa** – Ryhmään liittyvät henkilökunnan jäsenet.

- **lisaaHenkilokuntalainen**

```
public void lisaaHenkilokuntalainen( Henkilokuntalainen
henkilo )
```

- **Kuvaus**

Lisää ryhmään työläisen.

- **Parametrit**

* `henkilo` – lisättävä henkilökuntalainen

1.3 Luokka Ilpo

Class Ilpo

1.3.1 Kuvaus

```
public class Ilpo
extends javax.swing.JFrame
```

1.3.2 Metodien yhteenveto

```
getKurssiProperties() Palauttaa kurssin Propertiesit
getYleinenProperties() Palauttaa yleiset Propertiesit
laitaIkkunaPystyyn()
luoTietokantayhteys(String, String)
main(String[]) Main-metodi
setKurssiProperties(Properties) Asettaa kurssikohtaiset Propertiesit
```

1.3.3 Metodit

- **getKurssiProperties**

```
public static java.util.Properties getKurssiProperties(
)
```

- **Kuvaus**

Palauttaa kurssin Propertiesit

- **Palauttaa** – Kurssin Propertiesit

- **getYleinenProperties**

```
public static java.util.Properties
getYleinenProperties( )
```

- **Kuvaus**

Palauttaa yleiset Propertiesit

- **Palauttaa** – Yleiset Propertiesit

- **laitalkkunaPystyyn**

```
public void laitalkkunaPystyyn( )
```
- **luoTietokantayhteys**

```
public static void luoTietokantayhteys(  

  java.lang.String kayttajaTunnus, java.lang.String  

salasana ) heittää java.lang.Exception
```

 - **Parametrit**
 - * **kayttajaTunnus** –
 - * **salasana** –
- **main**

```
public static void main( java.lang.String[] args )
```

 - **Kuvaus**
Main-metodi
 - **Parametrit**
 - * **args** – main-metodin parametrit (eivät vaikuta mihinkään)
- **setKurssiProperties**

```
public static void setKurssiProperties(  

  java.util.Properties value )
```

 - **Kuvaus**
Asettaa kurssikohtaiset Propertiesit
 - **Parametrit**
 - * **value** – Kurssikohtaiset Propertiesit

1.4 Luokka IlpoRyhma

Ilpon sisäisen ryhmäjaon ryhmät.

1.4.1 Kuvaus

```
public class IlpoRyhma  

extends java.lang.Object  

implements java.io.Serializable
```

1.4.2 Konstruktorien yhteenveto

IlpoRyhma() Luo uuden IlpoRyhma-olion

1.4.3 Metodien yhteenveto

avaa() Avaa IlpoRyhma-olion lukituksen.

lisaaRyhmaListener(IlpoRyhmaListener) Lisää

IlmoRyhma-olioon sen tilaa kuuntelevan kuuntelijan.

lukitse() Asettaa IlpoRyhma-olion lukituksen.

muokkaaVapaatekstia(String) Muokkaa ryhmässä olevaa sen tilaa kuvaavaa tekstiä.

1.4.4 Konstruktorit

- **IlpoRyhma**

```
public IlpoRyhma ( )
```

- **Kuvaus**

Luo uuden IlpoRyhma-olion

1.4.5 Metodit

- **avaa**

```
public void avaa ( )
```

- **Kuvaus**

Avaa IlpoRyhma-olion lukituksen.

- **lisaaRyhmaListener**

```
public void lisaaRyhmaListener (
control.IlpoRyhmaListener r )
```

- **Kuvaus**

Lisää IlmoRyhma-olioon sen tilaa kuuntelevan kuuntelijan.

- **Parametrit**

* *r* – uusi IlpoRyhmaListener olio

- **lukitse**

```
public void lukitse ( )
```

- **Kuvaus**

Asettaa IlpoRyhma-olion lukituksen.

- **muokkaaVapaatekstia**

```
public void muokkaaVapaatekstia ( java.lang.String
uusiTeksti )
```

- **Kuvaus**

Muokkaa ryhmässä olevaa sen tilaa kuvaavaa tekstiä. Jos lisättävä teksti on tyhjä merkkijono asetetaan arvoksi null;

– Parametrit

* uusiTeksti – Tekstihuomio, joka lisätään ryhmään

1.5 Luokka Kurssi

Järjestelmän kannalta keskeinen tietorakenne, johon liittyy kaikki yhden kurssin ryhmittelyn kannalta olennainen tieto.

1.5.1 Kuvaus

```
public class Kurssi
extends java.lang.Object
implements java.io.Serializable
```

1.5.2 Konstruktorien yhteenveto

Kurssi(String, String, int, char, int, Tietokantayhteys) Luo tietokantakyselijän sekä parserin, joiden avulla noudetaan kannasta opiskelijoiden tiedot, ilmoittautumisen yhteydessä kerätyt tiedot, sekä kurssin pienryhmät.

1.5.3 Metodien yhteenveto

annaIlmoRyhmat() Palauttaa kurssiin liittyvät IlmoRyhma-oliot.

annaKayttajatunnukset() Palauttaa kokoelmaolion, joka sisältää kaikkien kurssille ilmoittautuneiden opiskelijoiden käyttäjätunnukset.

annaKurssiJarjestysnumero() Palauttaa arvonaan kurssin järjestysnumeron.

annaKurssikoodi() Palauttaa ryhmiteltävän kurssin kurssikoodin
annaKysymys(int) Palauttaa arvonaan sen Kysymys-olion, jossa on parametrin mukainen Ilpo-järjestysnumero.

annaLukukaudenNimi() Palauttaa ryhmiteltävän kurssin lukukauden nimen suomeksi

annaLukukausi() Palauttaa arvonaan lukukauden lyhenteen, jolloin kurssi luennoidaan.

annaLukuvuosi() Palauttaa lukuvuoden, jolloin ryhmiteltävä kurssin pidetään.

annaOpintojaksot() Palauttaa järjestelmän opintojaksot, toistaiseksi puhtaasti testikäytössä

annaOpiskelijanumerot() Palauttaa kokoelmaolion, joka sisältää kaikkien kurssille ilmoittautuneiden opiskelijoiden opiskelijanumerot.

- annaPuutteellisetOpiskelijat()** Palauttaa kokoelmaolion, joka koostuu niistä Opiskelija-olioista, joilla on puutteita opintosuorituksissa.
- annaRyhmatOpiskelijat()** Palauttaa kokoelmaolion, joka koostuu niistä Opiskelija-olioista, jotka eivät kuulu mihinkään IlpoRyhma-olioon.
- annaVapaaKysymys(int)** Palauttaa arvonaan sen vapaaKysymys-olion, jossa on parametrin mukainen Ilpo-järjestysnumero.
- lisaaKayttajatunnukset()** Lisää käyttäjätunnukset niille opiskelijoille, joilla niitä ei ole.
- lisaaKysymys(Kysymys)** Lisää Kurssi-olioon kysymyksen, johon on olemassa valmiit vastausvaihtoehdot.
- lisaaVapaakysymys(Vapaakysymys)** Lisää Kurssi-olioon kysymyksen, jonka vastaukset ovat vapaamuotoisia.
- noudaOpiskelijoidenOpintosuoritukset()** Noutaa opiskelijoiden opintosuoritukset kannasta ja luo Opiskelija-olioille suorituksia vastaavat SuoritettuOpintojakso-, Sivuaine- sekä SuoritettuSivuaineKurssi-oliot.
- paivitaIlmoRyhmatJaOpet()** Hakee tietokannasta kurssin pienryhmät sekä niiden ohjaajat ja luo vastaavat Henkilakuntalainen- sekä IlmoRyhmä-oliot.
- paivitaOpiskelijat()** Hakee tietokannasta kurssille ilmoittautuneiden opiskelijoiden tiedot, merkitse ilmoittautumisensa peruneet opiskelijat peruneiksi ja luo uusia ilmoittautuneita kohti uudet Opiskelija-oliot.
- paivitaOpiskelijoidenOpintosuoritukset()** Noutaa tietokannasta opiskelijoiden opintosuoritukset ja päivittää ne tarvittaessa.

1.5.4 Konstruktorit

- **Kurssi**

```
public Kurssi( java.lang.String nimi,
java.lang.String kurssikoodi, int lukuvuosi, char
lukukausi, int kurssiJarjestysnumero, db.Tietokantayhteys
yhteys )
```

- **Kuvaus**

Luo tietokantakyselijän sekä parserin, joiden avulla noudetaan kannasta opiskelijoiden tiedot, ilmoittautumisen yhteydessä kerätyt tiedot, sekä kurssin pienryhmät.

- **Parametrit**

- * nimi – kurssin nimi
- * kurssikoodi – Kurssin kurssikoodi

- * `lukuvuosi` – Lukuvuosi, jolloin kurssi luennoidaan
- * `lukukausi` – Lukukausi, jolloin kurssi luennoidaan
- * `kurssiJarjestysnumero` – Ilmaisee, kuinka mones kyseisen opintojakson ilmentymä kyseisellä
- * `yhteys` – Tietokantayhteys, joka annetaan kyselijälle lukukadella on kyseessä

1.5.5 Metodit

- **annaIlmoRyhmat**

```
public java.util.Collection annaIlmoRyhmat( )
```

- **Kuvaus**

Palauttaa kurssiin liittyvät IlmoRyhma-oliot.

- **Palauttaa** – Kurssin IlmoRyhma-oliot.

- **annaKayttajatunnukset**

```
public java.util.Collection annaKayttajatunnukset( )
```

- **Kuvaus**

Palauttaa kokoelmaolion, joka sisältää kaikkien kurssille ilmoittautuneiden opiskelijoiden käyttäjätunnukset.

- **Palauttaa** – Kokoelma, jossa opiskelijoiden käyttäjätunnukset

- **annaKurssiJarjestysnumero**

```
public int annaKurssiJarjestysnumero( )
```

- **Kuvaus**

Palauttaa arvonaan kurssin järjestysnumeron. Järjestysnumero kertoo monesko kyseisen opintojakson ilmentymä kurssi on kyseisellä lukukaudella.

- **Palauttaa** – Kurssin järjestysnumere lukukauden sisällä

- **annaKurssikoodi**

```
public java.lang.String annaKurssikoodi( )
```

- **Kuvaus**

Palauttaa ryhmiteltävän kurssin kurssikoodin

- **Palauttaa** – Kurssikoodi merkkijonona

- **annaKysymys**

```
public Kysymys annaKysymys( int ilponumero )
```

- **Kuvaus**

Palauttaa arvonaan sen Kysymys-olion, jossa on parametrin mukainen Ilpo-järjestysnumero.

- **Parametrit**

- * `ilponumero` – Järjestelmän sisäinen järjestysnumero kysymykselle.

- **Palauttaa** – Haluttu Kysymys-olio.

- **annaLukukaudenNimi**

```
public java.lang.String annaLukukaudenNimi( )
```

- **Kuvaus**

- Palauttaa ryhmiteltävän kurssin lukukauden nimen suomeksi

- **Palauttaa** – Merkkijono, joka on lukukauden nimi suomeksi

- **annaLukukausi**

```
public char annaLukukausi( )
```

- **Kuvaus**

- Palauttaa arvonaan lukukauden lyhenteen, jolloin kurssi luennoidaan.

- k = kevät, v = kesä, s = syksy

- **Palauttaa** – Lukuvuoden lyhenne

- **annaLukuvuosi**

```
public int annaLukuvuosi( )
```

- **Kuvaus**

- Palauttaa lukuvuoden, jolloin ryhmiteltävä kurssin pidetään.

- **Palauttaa** – Lukuvuosi kokonaislukuna

- **annaOpintojaksot**

```
public java.util.Hashtable annaOpintojaksot( )
```

- **Kuvaus**

- Palauttaa järjestelmän opintojaksot, toistaiseksi puhtaasti testikäytössä

- **annaOpiskelijanumerot**

```
public java.util.Collection annaOpiskelijanumerot( )
```

- **Kuvaus**

- Palauttaa kokoelmaolion, joka sisältää kaikkien kurssille

- ilmoittautuneiden opiskelijoiden opiskelijanumerot.

- **Palauttaa** – Kokoelma, jossa opiskelijoiden opiskelijanumerot

- **annaPuutteellisetOpiskelijat**

```
public java.util.Collection annaPuutteellisetOpiskelijat(  
)
```

- **Kuvaus**

Palauttaa kokoelmaolion, joka koostuu niistä Opiskelija-olioista, joilla on puutteita opintosuorituksissa.

- **Palauttaa** – Puutteliset Opiskelija-oliot.

- **annaRyhmittomatOpiskelijat**

```
public java.util.Collection annaRyhmittomatOpiskelijat(
)
```

- **Kuvaus**

Palauttaa kokoelmaolion, joka koostuu niistä Opiskelija-olioista, jotka eivät kuulu mihinkään IlpoRyhma-olioon. HTMLKirjoitin käyttää metodia. Niitä Opiskelija-olioita, joiden poistettuKurssilta-kentän arvo on true ei listata (Ilmoittautumisensa peruneet opiskelijat).

- **Palauttaa** – Ryhmittömät Opiskelija-oliot.

- **annaVapaaKysymys**

```
public VapaaKysymys annaVapaaKysymys( int ilponumero )
```

- **Kuvaus**

Palauttaa arvonaan sen vapaaKysymys-olion, jossa on parametrin mukainen Ilpo-järjestysnumero.

- **Parametrit**

* **ilponumero** – Järjestelmän sisäinen järjestysnumero kysymykselle.

- **Palauttaa** – Haluttu VapaaKysymys-olio.

- **lisaaKayttajatunnukset**

```
public void lisaaKayttajatunnukset( )
```

- **Kuvaus**

Lisää käyttäjätunnukset niille opiskelijoille, joilla niitä ei ole. Vain ne tunnukset, jotka löytyvät desmondista lisätään.

- **lisaaKysymys**

```
public void lisaaKysymys( Kysymys uusiKysymys )
```

- **Kuvaus**

Lisää Kurssi-olioon kysymyksen, johon on olemassa valmiit vastausvaihtoehdot.

- **Parametrit**

* **uusiKysymys** – Vaihtoehdollinen kysymys.

- **lisaaVapaakysymys**

```
public void lisaaVapaakysymys( Vapaakysymys
uusiVapaaKysymys )
```

- **Kuvaus**

Lisää Kurssi-olioon kysymyksen, jonka vastaukset ovat vapaamuotoisia.

- **Parametrit**

* `uusiVapaaKysymys` – parseoitu kysymys

- **noudaOpiskelijoidenOpintosuoritukset**

```
public void noudaOpiskelijoidenOpintosuoritukset( )
```

- **Kuvaus**

Noutaa opiskelijoiden opintosuoritukset kannasta ja luo Opiskelija-olioille suorituksia vastaavat SuoritettuOpintojakso-, Siuaine- sekä SuoritettuSiuaineKurssi-oliot.

- **paivitaIlmoRyhmatJaOpet**

```
public void paivitaIlmoRyhmatJaOpet( )
```

- **Kuvaus**

Hakee tietokannasta kurssin pienryhmät sekä niiden ohjaajat ja luo vastaavat Henkilakuntalainen- sekä IlmoRyhmä-oliot. Metodi ylikirjoittaa järjestelmässä olevat IlpoRyhmä- sekä Henkilokuntalainen-oliot sekä hävittää niihin mahdollisesti liittyvät viitteet.

- **paivitaOpiskelijat**

```
public void paivitaOpiskelijat( )
```

- **Kuvaus**

Hakee tietokannasta kurssille ilmoittautuneiden opiskelijoiden tiedot, merkitse ilmoittautumisensa peruneet opiskelijat peruneiksi ja luo uusia ilmoittautuneita kohti uudet Opiskelija-oliot.

- **paivitaOpiskelijoidenOpintosuoritukset**

```
public void paivitaOpiskelijoidenOpintosuoritukset( )
```

- **Kuvaus**

Noutaa tietokannasta opiskelijoiden opintosuoritukset ja päivittää ne tarvittaessa.

1.6 Luokka Kysymys

Ilmon monivalintakysymys ilpossa.

1.6.1 Kuvaus

```
public class Kysymys
extends java.lang.Object
implements java.io.Serializable
```

1.6.2 Konstruktorien yhteenveto

Kysymys(String, int, boolean) Luo uuden monivalintakysymys-olion.

1.6.3 Metodien yhteenveto

annaKysymysNumeroIlpessa() Palauttaa arvonaan kysymyksen järjestelmän sisäisen järjestysnumeron.

annaVastausvaihtoehto(String) Palauttaa vastauksen

lisaaVastausvaihtoehto(Vastausvaihtoehto) Ennaltamääritetyn vastausvaihtoehdon lisääminen

1.6.4 Konstruktorit

- **Kysymys**

```
public Kysymys( java.lang.String kysymys, int
kysymysNumeroIlpessa, boolean radioKysymys )
```

- **Kuvaus**

Luo uuden monivalintakysymys-olion.

- **Parametrit**

- * **kysymys** – Kysymys tekstimuodossa.

- * **kysymysNumeroIlpessa** – Järjestelmän sisäinen ID kysymykselle.

- * **radioKysymys** – Lippu, joka ilmoittaa onko kysymys radiobutton-tyyppinen.

1.6.5 Metodit

- **annaKysymysNumeroIlpessa**

```
public int annaKysymysNumeroIlpessa( )
```

- **Kuvaus**

Palauttaa arvonaan kysymyksen järjestelmän sisäisen järjestysnumeron. Järjestysnumero yksilöi kysymyksen.

- **Palauttaa** – Yksilöivä kysymyksen järjestysnumero.

- **annaVastausvaihtoehto**

```
public Vastausvaihtoehto annaVastausvaihtoehto (
    java.lang.String vastaus )
```

- **Kuvaus**

Palauttaa vastauksen

- **Parametrit**

* **vastaus** – tekstimuotainen vastaus

- **Palauttaa** – vastausvaihtoehto

- **lisaaVastausvaihtoehto**

```
public void lisaaVastausvaihtoehto( Vastausvaihtoehto
vaihtoehto )
```

- **Kuvaus**

Ennaltamääritetyn vastausvaihtoehdon lisääminen

- **Parametrit**

* **vaihtoehto** – vastausvaihtoehto

1.7 Luokka Opintojakso

TKTL:n kurssi, jonka opiskelija on saattanut suorittaa.

1.7.1 Kuvaus

```
public class Opintojakso
extends java.lang.Object
implements java.io.Serializable
```

1.7.2 Konstruktorien yhteenveto

Opintojakso(String, String, double) Luo uuden opintojakso-olion

1.7.3 Metodien yhteenveto

annaKurssikoodi() Palauttaa arvonaan opintojakson kurssikoodin.

lisaaSuoritettuOpintojakso(SuoritettuOpintojakso) Lisää

opintosuorituksen tämän opintojakson suoritukseen

toString() Palauttaa kirjallisen esityksen Opintojakso-oliosta.

1.7.4 Konstruktorit

- **Opintojakso**

```
public Opintojakso( java.lang.String nimi,
    java.lang.String kurssikoodi, double opintoviikot )
```

- **Kuvaus**

Luo uuden opintojakso-olion

- **Parametrit**

- * *nimi* – Opintojakson nimi

- * *kurssikoodi* – Opintojakson kurssikoodi

- * *opintoviikot* – Opintojakson laajuus opintoviikoissa

1.7.5 Metodit

- **annaKurssikoodi**

```
public java.lang.String annaKurssikoodi( )
```

- **Kuvaus**

Palauttaa arvonaan opintojakson kurssikoodin.

- **Palauttaa** – kurssikoodi merkkijonona.

- **lisaaSuoritettuOpintojakso**

```
public void lisaaSuoritettuOpintojakso(
    SuoritettuOpintojakso suoritus )
```

- **Kuvaus**

Lisää opintosuorituksen tämän opintojaksan suoritukseen

- **Parametrit**

- * *suoritus* – Opiskelijan kyseisen opintojakson suoritus

- **toString**

```
public java.lang.String toString( )
```

- **Kuvaus**

Palauttaa kirjallisen esityksen Opintojakso-oliosta.

- **Palauttaa** – Merkkijono, muotoa "kurssikoodi nimi (X ov)"

1.8 Luokka Opiskelija

Edustaa yhtä kurssille ilmoittautunutta opiskelijaa.

1.8.1 Kuvaus

public class Opiskelija
extends java.lang.Object
implements java.io.Serializable

1.8.2 Konstruktorien yhteenveto

Opiskelija(String, String, String, String, int, String, int, Tietokantakysely) Luo uuden Opiskelija-olion

1.8.3 Metodien yhteenveto

annaHetu() Palauttaa arvonaan opiskelijan henkilötunnuksen
annaKayttajatunnus() Palauttaa arvonaan opiskelijan käyttäjätunnuksen.
annaOpiskelijanumero() Palauttaa arvonaan opiskelijan opiskelijanumeron.
asettaKayttajatunnus(String) Asettaa opiskelijan käyttäjätunnuksen.
asettaOpiskelijanumero(String) Asettaa opiskelijan opiskelijanumeron.
lisaaOpiskelijaListener(OpiskelijaListener) Lisää Opiskelija-oliolle tämän tilaa kuuntelevan kuuntelijan.
lisaaSivuaineSuoritus(String, String, String, Date, String) Lisää suoritettun sivuainekurssin opiskelijan tietorakenteeseen.
lisaaSuoritettuOpintojakso(SuoritettuOpintojakso) Lisää SuoritettuOpintojakso-olion Opiskelija-olion tietoihin.
lisaaUupuvaOpintojakso(UupuvaOpintojakso) Luo uuden UupuvaOpintojakso-olion.
lisaaVapaavastaus(Vapaavastaus) Lisää opiskelijan vastaaman vapaavastauksen.
lisaaVastausvaihtoehto(Vastausvaihtoehto) Luo uuden Vastausvaihtoehto-olion.
muokkaaVapaatekstia(String) Muokkaa opiskelijassa olevaa hänen tilaansa kuvaavaa tekstiä.
nollaaSivuainesuoritukset() Tyhjentää opiskelijan sivuaineopinnot.
noudaOpintosuoritukset() Hakee kannasta opiskelijan opintosuoritukset ja alustaa opiskelijan opintosuoritukset-sisältävät tietorakenteet.
paivitaOpintosuoritukset() Päivittää opiskelijan Opintosuoritukset vastaamaan tietokannasta löytyviä.
peruillo() Merkitsee opiskelijan poistetuksi kurssilta.
peruutaLukitus() Mikäli Opiskelija-olio oli vain ryhmälukossa, poistaa ryhmälukon ja palauttaa Opiskelija-olion aktiiviseksi kaikkiin ryhmiin, joihin se on liitetty.

tarkastaEsitiedot() Tarkastaa Desmond.jar:n avulla, mitä ryhmiteltävälle kurssille vaadittavia kursseja opiskelijalta puuttuu.

teeOmaLukko(IlpoRyhma) Lukitsee Opiskelija-olion IlpoRyhma:ihin ja piilottaa opiskelijan muista ryhmistä, joihin hänet on liitetty.

teeRyhmaLukko(IlpoRyhma) Lukitsee Opiskelija ryhmälukolla annettuun IlpoRyhma-olioon.

toString() Tekstiesitys opiskelijasta

1.8.4 Konstruktorit

- **Opiskelija**

```
public Opiskelija( java.lang.String hetu,
java.lang.String opiskelijanumero, java.lang.String
nimi, java.lang.String sahkoposti, int aloitusvuosi,
java.lang.String paaaine, int ryhmaKannassa,
db.Tietokantakyselyija tietokantakyselyija )
```

- **Kuvaus**

Luo uuden Opiskelija-olion

- **Parametrit**

- * **hetu** – Opiskelijan henkilötunnus
- * **opiskelijanumero** – Opiskelijan opiskelijanumero
- * **nimi** – Opiskelijan nimi
- * **sahkoposti** – Opiskelijan sähköpostiosoite
- * **aloitusvuosi** – Vuosi, jolloin opiskelija on opintonsa aloittanut
- * **paaaine** – Opiskelijan pääaineen lyhenne
- * **ryhmaKannassa** – Ryhmä, johon opiskelija on sijoitettu ilmoittautumisen yhteydessä
- * **tietokantakyselyija** – Tietokantakyselyija, jota opiskelija käyttää jatkossa noutaakseen ilmoitettunsa ja opintosuoritteensa.

1.8.5 Metodit

- **annaHetu**

```
public java.lang.String annaHetu( )
```

- **Kuvaus**

Palauttaa arvonaan opiskelijan henkilötunnuksen

- **Palauttaa** – Opiskelijan henkilötunnus

- **annaKayttajatunnus**
`public java.lang.String annaKayttajatunnus()`
 - **Kuvaus**
Palauttaa arvonaan opiskelijan käyttäjätunnuksen.
 - **Palauttaa** – Opiskelijan käyttäjätunnus
- **annaOpiskelijanumero**
`public java.lang.String annaOpiskelijanumero()`
 - **Kuvaus**
Palauttaa arvonaan opiskelijan opiskelijanumeron.
 - **Palauttaa** – Opiskelijan opiskelijanumero
- **asetaKayttajatunnus**
`public void asetaKayttajatunnus(java.lang.String
uusiKayttajatunnus)`
 - **Kuvaus**
Asettaa opiskelijan käyttäjätunnuksen.
 - **Parametrit**
* `uusiKayttajatunnus` – uusi käyttäjätunnus
- **asetaOpiskelijanumero**
`public void asetaOpiskelijanumero(java.lang.String
uusiOpiskelijanumero)`
 - **Kuvaus**
Asettaa opiskelijan opiskelijanumeron.
 - **Parametrit**
* `uusiOpiskelijanumero` – tekstimuodossa oleva opiskelijanumero
- **lisaaOpiskelijaListener**
`public void lisaaOpiskelijaListener(
control.OpiskelijaListener o)`
 - **Kuvaus**
Lisää Opiskelija-oliolle tämän tilaa kuuntelevan kuuntelijan.
 - **Parametrit**
* `o` – Kuuntelija.
- **lisaaSivuaineSuoritus**
`public void lisaaSivuaineSuoritus(java.lang.String sid,
java.lang.String kk, java.lang.String n,
java.util.Date pvm, java.lang.String as)`

- **Kuvaus**

Lisää suoritettun sivuainekurssin opiskelijan tietorakenteeseen.

- **Parametrit**

- * `sid` – sivuaineen id Desmond järjestelmässä
- * `kk` – Kurssin kurssikoodi
- * `n` – Kurssin nimi
- * `pvm` – Päivämäärä, jolloin kurssi on suoritettu
- * `as` – Arvosana, mikä opiskelijalle on menty antamaan

- **lisaaSuoritettuOpintojakso**

```
public void lisaaSuoritettuOpintojakso (
    SuoritettuOpintojakso s )
```

- **Kuvaus**

Lisää SuoritettuOpintojakso-olion Opiskelija-olion tietoihin.

- **Parametrit**

- * `s` – SuoritettuOpintojakso-olio joka vastaa opiskelijan suoritusta.

- **lisaaUupuvaOpintojakso**

```
public void lisaaUupuvaOpintojakso ( UupuvaOpintojakso
    uoj )
```

- **Kuvaus**

Luo uuden UupuvaOpintojakso-olion.

- **Parametrit**

- * `uoj` – UupuvaOpintojakso-olio

- **lisaaVapaavastaus**

```
public void lisaaVapaavastaus ( Vapaavastaus v )
```

- **Kuvaus**

Lisää opiskelijan vastaaman vapaavastauksen.

- **Parametrit**

- * `v` – Vapaavastaus-olio

- **lisaaVastausvaihtoehto**

```
public void lisaaVastausvaihtoehto ( Vastausvaihtoehto v
    )
```

- **Kuvaus**

Luo uuden Vastausvaihtoehto-olion.

- **Parametrit**

- * `v` – Vastausvaihtoehto

- **muokkaaVapaatekstia**

```
public void muokkaaVapaatekstia ( java.lang.String  
uusiTeksti )
```

- **Kuvaus**

Muokkaa opiskelijassa olevaa hänen tilaansa kuvaavaa tekstiä. Jos lisättävä teksti on tyhjä merkkijono asetetaan arvoksi null;

- **Parametrit**

* uusiTeksti – Tekstihuomio, joka lisätään opiskelijaan

- **nollaaSivuainesuoritukset**

```
public void nollaaSivuainesuoritukset ( )
```

- **Kuvaus**

Tyhjentää opiskelijan sivuaineopinnot. Metodia käytetään sivuainesuoritusten yhteydessä.

- **noudaOpintosuoritukset**

```
public void noudaOpintosuoritukset ( )
```

- **Kuvaus**

Hakee kannasta opiskelijan opintosuoritukset ja alustaa opiskelijan opintosuoritukset-sisältävät tietorakenteet.

- **paivitaOpintosuoritukset**

```
public void paivitaOpintosuoritukset ( )
```

- **Kuvaus**

Päivittää opiskelijan Opintosuoritukset vastaamaan tietokannasta löytyviä.

- **perullmo**

```
public void perullmo ( )
```

- **Kuvaus**

Merkitsee opiskelijan poistetuksi kurssilta. Metodia käytetään tietokantapäivitysten yhteydessä.

- **peruutaLukitus**

```
public void peruutaLukitus ( )
```

- **Kuvaus**

Mikäli Opiskelija-olio oli vain ryhmälukossa, poistaa ryhmälukon ja palauttaa Opiskelija-olion aktiiviseksi kaikkiin ryhmiin, joihin se on liitetty. Mikäli Opiskelija-olio oli sekä ryhmälukittu että omalukittu, poistaa vain ryhmälukon ja jättää omalukon päälle. Mikäli Opiskelija-olio oli vain omalukittu, poistaa omalukon ja palauttaa Opiskelija-olion aktiiviseksi kaikkiin ryhmiin, joihin se on liitetty.

- **tarkastaEsitiedot**

```
public void tarkastaEsitiedot ( )
```

- **Kuvaus**

Tarkastaa Desmond.jar:n avulla, mitä ryhmiteltävälle kurssille vaadittavia kursseja opiskelijalta puuttuu.

- **teeOmaLukko**

```
public void teeOmaLukko ( IlpoRyhma i )
```

- **Kuvaus**

Lukitsee Opiskelija-olion IlpoRyhma i:hin ja piilottaa opiskelijan muista ryhmistä, joihin hänet on liitetty.

- **Parametrit**

* i – IlpoRyhma, johon opiskelija lukitaan.

- **teeRyhmaLukko**

```
public void teeRyhmaLukko ( IlpoRyhma i )
```

- **Kuvaus**

Lukitsee Opiskelija ryhmälukolla annettuun IlpoRyhma-olioon.

- **Parametrit**

* i – Ilporyhmä, johon opiskelija lukitaan. Mikäli Opiskelija on jo omalukittu johonkin ryhmään, tämän IlpoRyhmän täytyy olla sama, mihin Opiskelija on omalukittu.

- **toString**

```
public java.lang.String toString ( )
```

- **Kuvaus**

Tekstiesitys opiskelijasta

- **Palauttaa** – nimi hetu/opnro/käyttäjätunnus

1.9 Luokka Sivuaine

Opiskelijan yhden sivuaine-aineen suoritusten majapaikka.

1.9.1 Kuvaus

```
public class Sivuaine
extends java.lang.Object
implements java.io.Serializable
```

1.9.2 Konstruktorien yhteenveto

Sivuaine(String) Luodaan nimetön sivuaine

Sivuaine(String, String) Luodaan sivuaine, jonka nimi löytyi .properties tiedostosta

1.9.3 Metodien yhteenveto

lisaasuoritus(SuoritettuSivuainekurssi) Lisää opiskelijan suorittaman sivuainekurssin

1.9.4 Konstruktorit

- **Sivuaine**

```
public Sivuaine( java.lang.String id )
```

- **Kuvaus**

Luodaan nimetön sivuaine

- **Parametrit**

* **id** – Sivuaineen ID, peräisin Desmondista

- **Sivuaine**

```
public Sivuaine( java.lang.String sivuaineID,  
java.lang.String nimi )
```

- **Kuvaus**

Luodaan sivuaine, jonka nimi löytyi .properties tiedostosta

- **Parametrit**

* **sivuaineID** – Sivuaineen ID, peräisin Desmondista

* **nimi** – Sivuaineen nimi, jos sellainen on tiedossa

1.9.5 Metodit

- **lisaasuoritus**

```
public void lisaasuoritus( SuoritettuSivuainekurssi s  
)
```

- **Kuvaus**

Lisää opiskelijan suorittaman sivuainekurssin

- **Parametrit**

* **s** – Sivuainekurssi, joka on suoritettu

1.10 Luokka SuoritettuOpintojakso

Yhdistää opiskelijan ja tämän suorittaman tkt:n opintojakson toisiinsa. Pitää sisällään suorituksen tiedot (päivämäärä, arvosana)

1.10.1 Kuvaus

```
public class SuoritettuOpintojakso
extends java.lang.Object
implements java.io.Serializable
```

1.10.2 Konstruktorien yhteenveto

SuoritettuOpintojakso(Opintojakso, Opiskelija, Date, String)
Luo opintosuorituksen

1.10.3 Metodien yhteenveto

annaArvosana() Palauttaa arvonaan suorituksen arvosanan

annaKurssikoodi() Palauttaa arvonaan suoritettujen opintojaksojen kurssikoodin

annaPaivamaara() Palauttaa arvonaan suorituksen päivämäärän

vaihdaSuorittaja(Opiskelija) Vaihtaa kurssin suorittaneen Opiskelijan.

vertaaJaPaivita(SuoritettuOpintojakso) Vertailee kahta saman Opiskelija-olion opintosuoritusta.

1.10.4 Konstruktorit

- **SuoritettuOpintojakso**

```
public SuoritettuOpintojakso( Opintojakso opintojakso,
Opiskelija suorittaja, java.util.Date paivamaara,
java.lang.String arvosana )
```

- **Kuvaus**

Luo opintosuorituksen

- **Parametrit**

- * **opintojakso** – Suoritettu opintojakso
- * **suorittaja** – Sankarillisen suorituksen tehnyt soturisieluinen opiskelija
- * **paivamaara** – Suorituksen ajankohta
- * **arvosana** – Suorituksen arvosana

1.10.5 Metodit

- **annaArvosana**

```
public java.lang.String annaArvosana( )
```

- **Kuvaus**

Palauttaa arvonaan suorituksen arvosanan

- **Palauttaa** – Suorituksesta annettu arvosana.

- **annaKurssikoodi**

```
public java.lang.String annaKurssikoodi( )
```

- **Kuvaus**

Palauttaa arvonaan suoritettujen opintojaksojen kurssikoodin

- **Palauttaa** – Suoritettujen opintojaksojen kurssikoodi

- **annaPaivamaara**

```
public java.util.Date annaPaivamaara( )
```

- **Kuvaus**

Palauttaa arvonaan suorituksen päivämäärän

- **Palauttaa** – Päivämäärä, jolloin suoritus on tehty.

- **vaihdaSuorittaja**

```
public void vaihdaSuorittaja( Opiskelija suorittanut )
```

- **Kuvaus**

Vaihtaa kurssin suorittaneen Opiskelijan. Metodia käytetään opiskelijan opintosuorituksen päivittämisessä.

- **Parametrit**

* suorittanut – Opiskelija joka asetetaan suorittajaksi

- **vertaaJaPaivita**

```
public boolean vertaaJaPaivita( SuoritettuOpintojakso  
s )
```

- **Kuvaus**

Vertailee kahta saman Opiskelija-olion opintosuoritusta. Metodia käytetään opintosuoritusten päivitysten yhteydessä. Metodi päivittää suorituksen päivämäärän ja arvosanan, mikäli parametrin annettu (kannasta löytynyt) suoritus on uudempi.

- **Parametrit**

* s – Vertailtava opintosuoritus, jota ei muuteteta

- **Palauttaa** – True, mikäli suoritus viittaa samaan opintojaksoon.

1.11 Luokka SuoritettuSivuainekurssi

Opiskelijan suorittama sivuainekurssi.

1.11.1 Kuvaus

```
public class SuoritettuSivuainekurssi
extends java.lang.Object
implements java.io.Serializable
```

1.11.2 Konstruktorien yhteenveto

SuoritettuSivuainekurssi(String, String, Date, String) Luo uuden opiskelijan suorittaman sivuainekurssin.

1.11.3 Konstruktorit

- **SuoritettuSivuainekurssi**

```
public SuoritettuSivuainekurssi( java.lang.String
kurssikoodi, java.lang.String nimi, java.util.Date
paivamaara, java.lang.String arvosana )
```

- **Kuvaus**

Luo uuden opiskelijan suorittaman sivuainekurssin.

- **Parametrit**

- * **kurssikoodi** – Kurssin kurssikoodi
- * **nimi** – Kurssin nimi
- * **paivamaara** – Päivämäärä, jolloin kurssi on suoritettu
- * **arvosana** – Arvosana, mikä opiskelijalle on menty antamaan

1.12 Luokka UupuvaOpintojakso

Kurssin esitietoihin kuuluva opintojakso, jota opiskelija ei ole suorittanut. Tiedot pohjautuvat Desmondin tietokantaan.

1.12.1 Kuvaus

```
public class UupuvaOpintojakso
extends java.lang.Object
implements java.io.Serializable
```

1.12.2 Konstruktorien yhteenveto

UupuvaOpintojakso(Opintojakso) Luo uuden UupuvaOpintojakso-olion.

1.12.3 Metodien yhteenveto

vertaaSuoritukseen(SuoritettuOpintojakso) Vertaa UupuvaOpintosjakso-olion kurssikoodia SuoritettuOpintojakso-olion kurssikoodiin.

ylenKatso() Merkitsee uupuvan opintojakson sellaiseen tilaan, että puute ei ole näkyvillä.

1.12.4 Konstruktorit

- **UupuvaOpintojakso**

```
public UupuvaOpintojakso ( Opintojakso opintojakso )
```

- **Kuvaus**

Luo uuden UupuvaOpintojakso-olion. Puute on parametrina annettu kurssi.

- **Parametrit**

* opintojakso – puuttuva opintojakso

1.12.5 Metodit

- **vertaaSuoritukseen**

```
public boolean vertaaSuoritukseen (
    SuoritettuOpintojakso verrokki )
```

- **Kuvaus**

Vertaa UupuvaOpintosjakso-olion kurssikoodia SuoritettuOpintojakso-olion kurssikoodiin.

- **Parametrit**

* verrokki – SuoritettuOpintojakso-olio, joka saattaa vastatata uupuvaa.

- **Palauttaa** – True, mikäli kyse on samasta kurssista.

- **ylenKatso**

```
public void ylenKatso ( )
```

- **Kuvaus**

Merkitsee uupuvan opintojakson sellaiseen tilaan, että puute ei ole näkyvillä. Tämä tulee kyseeseen esimerkiksi silloin, kun opiskelija on suorittanut kurssin, eikä siitä ole vielä merkintää rekistereissä tai silloin, kun kurssi on korvattu muilla opinnoilla.

1.13 Luokka Vapaakysymys

Ilmoittatumisen yhteydessä kysytty kysymys, johon on voitu vastata vapaamuotoisesti.

1.13.1 Kuvaus

```
public class Vapaakysymys
extends java.lang.Object
implements java.io.Serializable
```

1.13.2 Konstruktorien yhteenveto

Vapaakysymys(String, int) Luo Vapaakysymys-olion

1.13.3 Metodien yhteenveto

annaKysymysNumeroIlpessa() Palauttaa arvonaan kysymyksen järjestelmän sisäisen järjestysnumeron.

1.13.4 Konstruktorit

- **Vapaakysymys**

```
public Vapaakysymys( java.lang.String kysymys, int
kysymysNumeroIlpessa )
```

- **Kuvaus**

Luo Vapaakysymys-olion

- **Parametrit**

- * *kysymys* – Kysytty asia

- * *kysymysNumeroIlpessa* – Ilpon sisäinen järjestysnumero kysymykselle, joka on määritetty kurssi.properties-tiedostossa.

1.13.5 Metodit

- **annaKysymysNumeroIlpessa**

```
public int annaKysymysNumeroIlpessa( )
```

- **Kuvaus**

Palauttaa arvonaan kysymyksen järjestelmän sisäisen järjestysnumeron. Järjestysnumero yksilöi kysymyksen.

- **Palauttaa** – Yksilöivä kysymyksen järjestysnumero.

1.14 Luokka Vapaavastaus

Opiskelijan vastaus vapaakysymykseen.

1.14.1 Kuvaus

```
public class Vapaavastaus
extends java.lang.Object
implements java.io.Serializable
```

1.14.2 Konstruktorien yhteenveto

Vapaavastaus(Vapaakysymys, String) Luo uuden Vapaavastaus-olion.

1.14.3 Konstruktorit

- **Vapaavastaus**

```
public Vapaavastaus( Vapaakysymys vapaakysymys,
java.lang.String vastaus )
```

- **Kuvaus**

Luo uuden Vapaavastaus-olion.

- **Parametrit**

- * `vastaus` – Kysymys johon on vastattu.
- * `vapaakysymys` – Vastaus joka on annettu Ilmoon.

1.15 Luokka Vastausvaihtoehto

Vastausvaihtoehto sellaisen kysymyksen, johon on valmiit vastausvaihtoehdot.

1.15.1 Kuvaus

```
public class Vastausvaihtoehto
extends java.lang.Object
implements java.io.Serializable
```

1.15.2 Konstruktorien yhteenveto

Vastausvaihtoehto(String, String, Kysymys) Luo uuden vastausvaihtoehdon.

1.15.3 Metodien yhteenveto

annaVastausnumero() Palauttaa arvonaan vastauksen numeron Ilmo-lomakkeessa.

lisaaVastannutOpiskelija(Opiskelija) Lisää yhden juuri näin tähän kysymykseen vastanneen opiskelijan vastausvaihtoehdon tietorakenteeseen.

1.15.4 Konstruktorit

- **Vastausvaihtoehto**

```
public Vastausvaihtoehto( java.lang.String vaihtoehto,
java.lang.String vastausnumero, Kysymys kysymys )
```

- **Kuvaus**

Luo uuden vastausvaihtoehdon.

- **Parametrit**

- * **vaihtoehto** – Vastaus tekstimuodossa
- * **vastausnumero** – Vastauksen numero ilmo-lomakkeella
- * **kysymys** – Kysymys, johon vastausvaihtoehto liittyy

1.15.5 Metodit

- **annaVastausnumero**

```
public java.lang.String annaVastausnumero( )
```

- **Kuvaus**

Palauttaa arvonaan vastauksen numeron Ilmo-lomakkeessa.

- **Palauttaa** – Vastauksen numero Ilmo-lomakkeessa

- **lisaaVastannutOpiskelija**

```
public void lisaaVastannutOpiskelija( Opiskelija vastannut
)
```

- **Kuvaus**

Lisää yhden juuri näin tähän kysymykseen vastanneen opiskelijan vastausvaihtoehdon tietorakenteeseen.

- **Parametrit**

- * **vastannut** – Opiskelija, joka on erehtynyt vastaamaan kysymykseen vaihtoehdon mukaisesti

Chapter 2

Pakkaus fi.helsinki.cs.ilpo.control

<i>Pakkaus Sisältö</i>	<i>Sivu</i>
Rajapinnat	
IlmoRyhmaListener	31
IlmoRyhmä-olion kuuntelija.	
IlpoRyhmaListener	32
IlpoRyhmä-olion kuuntelija.	
NakymaRyhmaHKListener	32
PieniHKBoxi-olentoon tapahtuvia tapahtumia kuunteleva kuuntelija.	
NakymaRyhmaListener	32
NakymaRyhmä-olioita kuunteleva rajapinta.	
OpiskelijaListener	33
Opiskelija-oliota kuunteleva rajapinta.	
PalloListener	33
Kuuntelee Pallo-olioihin kohdistuvia toimintoja.	
Luokat	
KontrolliPalikka	34
KontrolliPalikka kuvaa kaikki kontrolliluokkia, eikä sitä ole oikeasti olemassa.	
KontrolliYlenkatsoUupuva	34
Suorittaa uupuvien opintosuoritusten ylenkatsomisen.	

2.1 Luokka IlmoRyhmaListener

IlmoRyhmä-olion kuuntelija. Kuuntelee IlmoRyhmä-oliossa tapahtuvia muutoksia.

2.1.1 Kuvaus

```
public interface IlmoRyhmaListener
```

2.1.2 Kaikki tunnetut alirajapinnat

HKBoxi (in 3.3, page 39), IsoHKBoxi (in 3.5, page 40), PieniHKBoxi (in 3.13, page 45)

2.1.3 Kaikki luokat, joiden tiedetään toteuttavan rajapinnan

HKBoxi (in 3.3, page 39)

2.2 Luokka IlpoRyhmaListener

IlpoRyhmä-olion kuuntelija. Kuuntelee IlpoRyhmä-oliossa tapahtuvia muutoksia.

2.2.1 Kuvaus

public interface IlpoRyhmaListener

2.2.2 Kaikki tunnetut alirajapinnat

NakymaRyhma (in 3.7, page 41), NakymaUusiRyhma (in 3.8, page 42),
NakymaVanhaRyhma (in 3.9, page 42)

2.2.3 Kaikki luokat, joiden tiedetään toteuttavan rajapinnan

NakymaRyhma (in 3.7, page 41)

2.3 Luokka NakymaRyhmaHKListener

PieniHKBoxi-olentoon tapahtuvia tapahtumia kuunteleva kuuntelija.

2.3.1 Kuvaus

public interface NakymaRyhmaHKListener

2.3.2 Kaikki tunnetut alirajapinnat

KontrolliPalikka (in 2.7, page 34)

2.3.3 Kaikki luokat, joiden tiedetään toteuttavan rajapinnan

KontrolliPalikka (in 2.7, page 34)

2.4 Luokka NakymaRyhmaListener

NakymaRyhma-olioita kuunteleva rajapinta.

2.4.1 Kuvaus

public interface NakymaRyhmaListener

2.4.2 Kaikki tunnetut alirajapinnat

KontrolliPalikka (in 2.7, page 34)

2.4.3 Kaikki luokat, joiden tiedetään toteuttavan rajapinnan

KontrolliPalikka (in 2.7, page 34)

2.5 Luokka OpiskelijaListener

Opiskelija-oliota kuunteleva rajapinta. Kuuntelee Opiskelija-oliossa tapahtuvia muutoksia.

2.5.1 Kuvaus

public interface OpiskelijaListener

2.5.2 Kaikki tunnetut alirajapinnat

KopioituvaPallo (in 3.6, page 41), Pallo (in 3.12, page 45), SiirtyvaPallo (in 3.19, page 48)

2.5.3 Kaikki luokat, joiden tiedetään toteuttavan rajapinnan

Pallo (in 3.12, page 45)

2.6 Luokka PalloListener

Kuuntelee Pallo-olioihin kohdistuvia toimintoja.

2.6.1 Kuvaus

public interface PalloListener

2.6.2 Kaikki tunnetut alirajapinnat

KontrolliPalikka (in 2.7, page 34)

2.6.3 Kaikki luokat, joiden tiedetään toteuttavan rajapinnan

KontrolliPalikka (in 2.7, page 34)

2.7 Luokka KontrolliPalikka

KontrolliPalikka kuvaa kaikki kontrolliluokkia, eikä sitä ole oikeasti olemassa. Kontrolliluokat nimetään oikeasti tyyliin "KontrolliToiminto", missä Toiminto on suoritettavan toiminnon nimi. Toteutettavat Listener- rajapinnatkin ovat jokaisella kontrolliluokalla omat. Kaikki eivät siis toteuta kaikkia kuuntelijoita. Kontrolliluokat kuuntelevat siis käyttöliittymän komponentteja (view-osaa) ja ilmoittavat halutusta toiminnasta sisäiselle tietorakenteelle (model-osaan). Esimerkiksi KontrolliYlenkatsoUupuva suorittaa uupuvien opintosuoritusten ylenkatsomisen.

2.7.1 Kuvaus

```
public class KontrolliPalikka
extends java.lang.Object
implements PalloListener, NakymaRyhmaListener, NakymaRyhmaHKListener
```

2.7.2 Konstruktorien yhteenveto

```
KontrolliPalikka()
```

2.7.3 Konstruktorit

- **KontrolliPalikka**

```
public KontrolliPalikka( )
```

2.8 Luokka KontrolliYlenkatsoUupuva

Suorittaa uupuvien opintosuoritusten ylenkatsomisen. Tämä luokka on suunnitteluvaiheessa esimerkkiluokka kontrolliluokista, joita myös KontrolliPalikka-luokka kuvaa.

2.8.1 Kuvaus

```
public class KontrolliYlenkatsoUupuva
extends java.lang.Object
implements java.awt.event.ActionListener
```

2.8.2 Konstruktorien yhteenveto

KontrolliYlenkatsoUupuva()

2.8.3 Metodien yhteenveto

actionPerformed(ActionEvent) Tavallinen actionPerformed-metodi.

2.8.4 Konstruktorit

- **KontrolliYlenkatsoUupuva**
`public KontrolliYlenkatsoUupuva ()`

2.8.5 Metodit

- **actionPerformed**
`public void actionPerformed (
java.awt.event.ActionEvent event)`

- **Kuvaus**

Tavallinen actionPerformed-metodi. Suoritetaan, kun se, mitä tällä kuunnellaan, päättää kutsua tätä metodia.

- **Parametrit**

* `event` – ActionEvent, joka laukaisi tämän Listenerin.

Chapter 3

Pakkaus fi.helsinki.cs.ilpo.view

<i>Pakkaus Sisältö</i>	<i>Sivu</i>
Luokat	
AutentikointiIkkuna	37
Class AutentikointiIkkuna	
HenkilotietoSolu	38
Näytön vasempaan alanurkkaan tietosoluun asetettava opiskelijan tarkemmat tiedot sisältävä näkymä.	
HKBoxi	39
Ilmoon talletetun ryhmän henkilökuntaa kuvaavan graafinen elementin yliluokka.	
HKSortSolu	40
Ilmosta saatavien ryhmien henkilökuntaa esittäviä palikoita sisältävän listan sisältävä solu näytön vasempaan ylänurkkaan.	
IsoHKBoxi	40
Vasemmassa ylänurkassa näkyvä Ilmon ryhmähenkilökuntaa esittävä graafinen elementti.	
KopioituvaPallo	41
Vasemman ylänurkan sortsolun opiskelijasortsolun opiskelijaa esittävä graafinen pallo, joka voidaan vetää ryhmänäkymän puolelle oikealle puolelle näyttöä johonkin ryhmään, jolloin opiskelija liitetään kyseiseen ryhmään, mutta alkuperäinen pallo jää paikalleen.	
NakymaRyhma	41
Ryhmää esittävä graafinen representaatio asetettavaksi näytön oikeaan puoliskoon, ryhmäsoluun.	
NakymaUusiRyhma	42
NakymaUusiRyhma on ryhmasolun vasempaan alanurkkaan luotava tyhjä ryhmä, joka opiskelijoita lisättäessä korvataan NakymaVanhaRyhmallä.	
NakymaVanhaRyhma	42
Näytön oikeaan puoliskoon, ryhmäsoluun, liitettävä ryhmän representaatio, johon lisätty opiskelijoita, ja joka näin liittyy johonkin ilporyhmään.	
OpiskelijaListassaJComponent	43

Tietosolun ollessa listanäkymässä, ovat listanäkymän näkymät `OpiskelijaListassaJ-Componentteja`.

OpiskelijaSortSolu	44
Ikkunan vasempaan ylänurkkaan Sortsoluun asetettava opiskelijapalloja sisältävä näkymä.	
Pallo	45
Järjestelmän peruselementti, opiskelijaa esittävä pallo, jonka ulkoasun perivät ainakin <code>KopioituvaPallo</code> sekä <code>SiirtyvaPallo</code> .	
PieniHKBoxi	45
Oikealla ryhmänäytössä ryhmän sisällä sijaitseva graafinen elementti, joka kuvaa ilmoryhmän henkilökuntaa, joka on kiinnitetty ilporyhmään.	
PoistaOpiskelijaButton	46
Button, joka poistaa opiskelijan.	
PoistaRyhmaButton	46
Button, jota painamalla ryhmä poistuu.	
PoistaRyhmostaButton	47
Button, jota painamalla <code>Opiskelija</code> poistuu <code>IlpoRyhmästä</code> .	
RyhmaSolu	47
Näytön oikea puoli, jossa ilpon ryhmät kelluvat.	
RyhmätietoSolu	48
Valitun ryhmän opiskelijoiden tiedot listassa näytävä näkymä, joka sijaitsee tietosolussa näytän vasemmassa alanurkassa.	
SiirtyvaPallo	48
Ryhmasolun ryhmissä kelluvat opiskelijoita kuvaavat pallot, jotka siirrettäessä siirtyvät uuteen ryhmään.	
SortSolu	49
Vasemman ylänurkan solu, johon asetetaan erilaisia järjestämiseen liittyviä soluja.	
TietoSolu	50
Näytön vasemmassa alanurkassa oleva solu, johon asetetaan joko opiskelijan tarkempia tietoja sisältävä solu tai vaihtoehtoisesti valittujen opiskelijoiden tiivistettyjä tietoja sisältävä lista.	
TulosRyhmanakyma	50
Vasemman ylänurkan järjestettyjä opiskelijapalloja sisältävä näkymä.	
YlenkatsoJButton	51
Button, jota painamalla käyttäjä voi ylenkatsoa uupuvan opintosuorituksen.	

3.1 Luokka `AutentikointiIkkuna`

Class `AutentikointiIkkuna`

3.1.1 Kuvaus

```
public class AutentikointiIkkuna
```



```
extends javax.swing.JFrame
```

3.1.2 Kenttien yhteenveto

```
passwordField  
userNameField
```

3.1.3 Konstruktorien yhteenveto

```
AutentikointiIkkuna()
```

3.1.4 Metodien yhteenveto

```
main(String[])  
onkoAutentikointiOK()  
onkoPainettuOk()
```

3.1.5 Kentät

- protected javax.swing.JTextField **userNameField**
- protected javax.swing.JPasswordField **passwordField**

3.1.6 Konstruktorit

- **AutentikointiIkkuna**
public **AutentikointiIkkuna**()

3.1.7 Metodit

- **main**
public static void **main**(java.lang.String[] **args**)
- **onkoAutentikointiOK**
public boolean **onkoAutentikointiOK**()
- **onkoPainettuOk**
public boolean **onkoPainettuOk**()

3.2 Luokka HenkilotietoSolu

Näytön vasempaan alanurkkaan tietosoluun asetettava opiskelijan tarkemmat tiedot sisältävä näkymä.

3.2.1 Kuvaus

```
public class HenkilotietoSolu
extends javax.swing.JPanel
```

3.2.2 Konstruktorien yhteenveto

HenkilotietoSolu() Luodaan TietoSoluun liittyvä opiskelijatietonäkymä.

3.2.3 Konstruktorit

- **HenkilotietoSolu**

```
public HenkilotietoSolu ( )
```

- **Kuvaus**

Luodaan TietoSoluun liittyvä opiskelijatietonäkymä.

3.3 Luokka HKBoxi

Ilmoon talletetun ryhmän henkilökuntaa kuvaavan graafinen elementin ylikuokka.

3.3.1 Kuvaus

```
public class HKBoxi
extends java.lang.Object
implements fi.helsinki.cs.ilpo.control.IlmoRyhmaListener
```

3.3.2 Kaikki tunnetut aliluokat

IsoHKBoxi (in 3.5, page 40), PieniHKBoxi (in 3.13, page 45)

3.3.3 Konstruktorien yhteenveto

HKBoxi(IlmoRyhma) Luo uuden Ilmoryhmään liittyvän HKBoxin

3.3.4 Konstruktorit

- **HKBoxi**

```
public HKBoxi( fi.helsinki.cs.ilpo.IlmoRyhma
 ilmoRyhma )
```

- **Kuvaus**

Luo uuden Ilmoryhmään liittyvän HKBoxin

- **Parametrit**

* ilmoRyhma – elementin kuvaama ryhmähenkilökunta

3.4 Luokka HKSortSolu

Ilmosta saatavien ryhmien henkilökuntaa esittäviä palikoita sisältävän listan sisältävä solu näytön vasempaan ylänurkkaan.

3.4.1 Kuvaus

```
public class HKSortSolu
extends javax.swing.JPanel
```

3.4.2 Konstruktorien yhteenveto

HKSortSolu() Luodaan HKSortSolu

3.4.3 Konstruktorit

- **HKSortSolu**
`public HKSortSolu ()`
 - **Kuvaus**
Luodaan HKSortSolu

3.5 Luokka IsoHKBoxi

Vasemmassa ylänurkassa näkyvä Ilmon ryhmähenkilökuntaa esittävä graafinen elementti.

3.5.1 Kuvaus

```
public class IsoHKBoxi
extends fi.helsinki.cs.ilpo.view.HKBoxi (in 3.3, page 39)
```

3.5.2 Konstruktorien yhteenveto

IsoHKBoxi(IlmoRyhma) Luo uuden Ilmoryhmän henkilökuntaa esittävän kuvajaisen.

3.5.3 Konstruktorit

- **IsoHKBoxi**
`public IsoHKBoxi(fi.helsinki.cs.ilpo.IlmoRyhma ilmoRyhma)`
 - **Kuvaus**
Luo uuden Ilmoryhmän henkilökuntaa esittävän kuvajaisen.

– Parametrit

* ilmoRyhma – elementin kuvaama ryhmähenkilökunta

3.6 Luokka KopioituvaPallo

Vasemman ylänurkan sortsolun opiskelijasortsolun opiskelijaa esittävä graafinen pallo, joka voidaan vetää ryhmänäkymän puolelle oikealle puolelle näyttöä johonkin ryhmään, jolloin opiskelija liitetään kyseiseen ryhmään, mutta alkuperäinen pallo jää paikalleen.

3.6.1 Kuvaus

```
public class KopioituvaPallo
extends fi.helsinki.cs.ilpo.view.Pallo (in 3.12, page 45)
```

3.6.2 Konstruktorien yhteenveto

KopioituvaPallo(Opiskelija) Luodaan sortsolussa asuva näkyvä pallo.

3.6.3 Konstruktorit

- **KopioituvaPallo**

```
public KopioituvaPallo( fi.helsinki.cs.ilpo.Opiskelija
opiskelija )
```

 - **Kuvaus**
Luodaan sortsolussa asuva näkyvä pallo.
 - **Parametrit**
* opiskelija – Opiskelija, jota pallo esittää.

3.7 Luokka NakymaRyhma

Ryhmää esittävä graafinen representaatio asetettavaksi näytön oikeaan puoliskoon, ryhmäsoluun.

3.7.1 Kuvaus

```
public abstract class NakymaRyhma
extends java.lang.Object
implements fi.helsinki.cs.ilpo.control.IlpoRyhmaListener
```

3.7.2 Kaikki tunnetut aliluokat

NakymaUusiRyhma (in 3.8, page 42), NakymaVanhaRyhma (in 3.9, page 42)

3.7.3 Konstruktorien yhteenveto

NakymaRyhma() Tehdään NakymaUusiRyhman ja NakymaVanhaRyhman yhteiset konstruktointitoiminnot.

3.7.4 Konstruktorit

- **NakymaRyhma**

```
public NakymaRyhma( )
```

- **Kuvaus**

Tehdään NakymaUusiRyhman ja NakymaVanhaRyhman yhteiset konstruktointitoiminnot.

3.8 Luokka NakymaUusiRyhma

NakymaUusiRyhma on ryhmasolun vasempaan alanurkkaan luotava tyhjä ryhmä, joka opiskelijoita lisättäessä korvataan NakymaVanhaRyhmallä.

3.8.1 Kuvaus

```
public class NakymaUusiRyhma  
extends fi.helsinki.cs.ilpo.view.NakymaRyhma (in 3.7, page 41)
```

3.8.2 Konstruktorien yhteenveto

NakymaUusiRyhma()

3.8.3 Konstruktorit

- **NakymaUusiRyhma**

```
public NakymaUusiRyhma( )
```

3.9 Luokka NakymaVanhaRyhma

Näytön oikeaan puoliskoon, ryhmäsoluun, liitettävä ryhmän representaatio, johon lisätty opiskelijoita, ja joka näin liittyy johonkin ilporyhmään.

3.9.1 Kuvaus

public class NakymaVanhaRyhma
extends fi.helsinki.cs.ilpo.view.NakymaRyhma (in 3.7, page 41)

3.9.2 Konstruktorien yhteenveto

NakymaVanhaRyhma(IlpoRyhma) Luodaan ilpoRyhmän graafinen ilmentymä

3.9.3 Konstruktorit

- **NakymaVanhaRyhma**
 public **NakymaVanhaRyhma** (
 fi.helsinki.cs.ilpo.IlpoRyhma **ilpoRyhma**)
 - **Kuvaus**
 Luodaan ilpoRyhmän graafinen ilmentymä
 - **Parametrit**
 * **ilpoRyhma** – NakymaRyhman esittämä ryhmä

3.10 Luokka OpiskelijaListassaJComponent

Tietosolun ollessa listanäkymässä, ovat listanäkymän näkymät OpiskelijaListassaJComponentteja.

3.10.1 Kuvaus

public class OpiskelijaListassaJComponent
extends javax.swing.JComponent

3.10.2 Konstruktorien yhteenveto

OpiskelijaListassaJComponent(Opiskelija) Luodaan ryhmätietosoluun liitettävä komponentti.

3.10.3 Konstruktorit

- **OpiskelijaListassaJComponent**
 public **OpiskelijaListassaJComponent** (
 fi.helsinki.cs.ilpo.Opiskelija **opiskelija**)
 - **Kuvaus**
 Luodaan ryhmätietosoluun liitettävä komponentti.

3.11 Luokka OpiskelijaSortSolu

Ikkunan vasempaan ylänurkkaan Sortsoluun asetettava opiskelijapalloja sisältävä näkymä.

3.11.1 Kuvaus

```
public class OpiskelijaSortSolu
extends javax.swing.JPanel
```

3.11.2 Konstruktorien yhteenveto

OpiskelijaSortSolu() Luodaan ilmentymä Opiskelijasortsolusta

3.11.3 Metodien yhteenveto

addTulosRyhmaNakyma(TulosRyhmaNakyma) Lisätään
TulosRyhmaNakyma järjestämisen yhteydessä näkymään.
paintComponent(Graphics)

3.11.4 Konstruktorit

- **OpiskelijaSortSolu**
`public OpiskelijaSortSolu ()`
 - **Kuvaus**
Luodaan ilmentymä Opiskelijasortsolusta

3.11.5 Metodit

- **addTulosRyhmaNakyma**
`public void addTulosRyhmaNakyma (TulosRyhmaNakyma
tulosRyhmaNakyma)`
 - **Kuvaus**
Lisätään TulosRyhmaNakyma järjestämisen yhteydessä näkymään.
 - **Parametrit**
* `tulosRyhmaNakyma` – Lisättävä TulosRyhmaNakyma
- **paintComponent**
`protected void paintComponent (java.awt.Graphics arg0
)`

3.12 Luokka Pallo

Järjestelmän peruselementti, opiskelijaa esittävä pallo, jonka ulkoasun perivät ainakin KopioituvaPallo sekä SiirtyvaPallo.

3.12.1 Kuvaus

```
public abstract class Pallo
extends java.lang.Object
implements fi.helsinki.cs.ilpo.control.OpiskelijaListener
```

3.12.2 Kaikki tunnetut aliluokat

KopioituvaPallo (in 3.6, page 41), SiirtyvaPallo (in 3.19, page 48)

3.12.3 Konstruktorien yhteenveto

Pallo() Luodaan opiskelijaa esittävä graafinen pallo.

3.12.4 Konstruktorit

- **Pallo**
`public Pallo()`

- **Kuvaus**

Luodaan opiskelijaa esittävä graafinen pallo.

3.13 Luokka PieniHKBoxi

Oikealla ryhmänäytössä ryhmän sisällä sijaitseva graafinen elementti, joka kuvaa Ilmoryhmän henkilökuntaa, joka on kiinnitetty ilporyhmään.

3.13.1 Kuvaus

```
public class PieniHKBoxi
extends fi.helsinki.cs.ilpo.view.HKBoxi (in 3.3, page 39)
```

3.13.2 Konstruktorien yhteenveto

PieniHKBoxi(IlmoRyhma) Luo uuden Ilmoryhmän henkilökuntaa esittävän kuvajaisen.

3.13.3 Konstruktorit

- **PieniHKBoxi**

```
public PieniHKBoxi( fi.helsinki.cs.ilpo.IlmoRyhma
  ilmoRyhma )
```

- **Kuvaus**

Luo uuden Ilmoryhmän henkilökuntaa esittävän kuvajaisen.

- **Parametrit**

* ilmoRyhma – elementin kuvaama ryhmähenkilökunta

3.14 Luokka PoistaOpiskelijaButton

Button, joka poistaa opiskelijan. Jostain.

3.14.1 Kuvaus

```
public class PoistaOpiskelijaButton
  extends javax.swing.JButton
```

3.14.2 Konstruktorien yhteenveto

PoistaOpiskelijaButton() Luo tämän olion.

3.14.3 Konstruktorit

- **PoistaOpiskelijaButton**

```
public PoistaOpiskelijaButton( )
```

- **Kuvaus**

Luo tämän olion.

3.15 Luokka PoistaRyhmaButton

Button, jota painamalla ryhmä poistuu.

3.15.1 Kuvaus

```
public class PoistaRyhmaButton
  extends javax.swing.JButton
```

3.15.2 Konstruktorien yhteenveto

PoistaRyhmaButton()

3.15.3 Konstruktorit

- **PoistaRyhmaButton**
`public PoistaRyhmaButton ()`

3.16 Luokka PoistaRyhmasterButton

Button, jota painamalla Opiskelija poistuu IlpoRyhmästä.

3.16.1 Kuvaus

```
public class PoistaRyhmasterButton
extends javax.swing.JButton
```

3.16.2 Konstruktorien yhteenveto

```
PoistaRyhmasterButton()
```

3.16.3 Konstruktorit

- **PoistaRyhmasterButton**
`public PoistaRyhmasterButton ()`

3.17 Luokka Ryhmasolu

Näytön oikea puoli, jossa ilpon ryhmät kelluvat.

3.17.1 Kuvaus

```
public class Ryhmasolu
extends javax.swing.JPanel
```

3.17.2 Konstruktorien yhteenveto

```
Ryhmasolu() Luodaan ilpoon asetettava ryhmänäkymä.
```

3.17.3 Metodien yhteenveto

```
paintComponent(Graphics)
```

3.17.4 Konstruktorit

- **RyhmaSolu**

```
public RyhmaSolu ( )
```

- **Kuvaus**

Luodaan ilpoon asetettava ryhmänäkymä.

3.17.5 Metodit

- **paintComponent**

```
protected void paintComponent ( java.awt.Graphics arg0  
)
```

3.18 Luokka RyhmatietoSolu

Valitun ryhmän opiskelijoiden tiedot listassa näytävä näkymä, joka sijaitsee tietosolussa näytän vasemmassa alanurkassa.

3.18.1 Kuvaus

```
public class RyhmatietoSolu  
extends java.lang.Object
```

3.18.2 Konstruktorien yhteenveto

RyhmatietoSolu() Luodaan ryhmätietosolu valitun ryhmän
Opiskelijoiden pohjalta

3.18.3 Konstruktorit

- **RyhmatietoSolu**

```
public RyhmatietoSolu ( )
```

- **Kuvaus**

Luodaan ryhmätietosolu valitun ryhmän Opiskelijoiden pohjalta

3.19 Luokka SiirtyvaPallo

Ryhmasolun ryhmissä kelluvat opiskelijoita kuvaavat pallot, jotka siirrettäessä siirtyvät uuteen ryhmään.

3.19.1 Kuvaus

public class SiirtyvaPallo
extends fi.helsinki.cs.ilpo.view.Pallo (in 3.12, page 45)

3.19.2 Konstruktorien yhteenveto

SiirtyvaPallo(Opiskelija) Luodaan NakymaRyhmaan siirrettäessä näkyvä pallo.

3.19.3 Konstruktorit

- **SiirtyvaPallo**
public **SiirtyvaPallo**(fi.helsinki.cs.ilpo.Opiskelija opiskelija)
 - **Kuvaus**
Luodaan NakymaRyhmaan siirrettäessä näkyvä pallo.
 - **Parametrit**
* opiskelija – Opiskelija, jota pallo esittää.

3.20 Luokka SortSolu

Vasemman ylänurkan solu, johon asetetaan erilaisia järjestämiseen liittyviä soluja.

3.20.1 Kuvaus

public class SortSolu
extends java.lang.Object

3.20.2 Konstruktorien yhteenveto

SortSolu() Luo päänäyttöön liitettävän järjestyssolun.

3.20.3 Konstruktorit

- **SortSolu**
public **SortSolu**()
 - **Kuvaus**
Luo päänäyttöön liitettävän järjestyssolun.

3.21 Luokka TietoSolu

Näytön vasemmassa alanurkassa oleva solu, johon asetetaan joko opiskelijan tarkempia tietoja sisältävä solu tai vaihtoehtoisesti valittujen opiskelijoiden tiivistettyjä tietoja sisältävä lista.

3.21.1 Kuvaus

```
public class TietoSolu
extends java.lang.Object
```

3.21.2 Konstruktorien yhteenveto

TietoSolu() Luodaan päänäyttöön kiinnitettävä TietoSolu.

3.21.3 Konstruktorit

- **TietoSolu**

```
public TietoSolu ( )
```

- **Kuvaus**

Luodaan päänäyttöön kiinnitettävä TietoSolu.

3.22 Luokka TulosRyhmaNakyma

Vasemman ylänurkan järjestettyjä opiskelijapalloja sisältävä näkymä.

3.22.1 Kuvaus

```
public class TulosRyhmaNakyma
extends java.lang.Object
```

3.22.2 Konstruktorien yhteenveto

TulosRyhmaNakyma() Luodaan sortsoluun liittyvä järjestysnäkö.

3.22.3 Konstruktorit

- **TulosRyhmaNakyma**

```
public TulosRyhmaNakyma ( )
```

- **Kuvaus**

Luodaan sortsoluun liittyvä järjestysnäkö.

3.23 Luokka YlenkatsoJButton

Button, jota painamalla käyttäjä voi ylenkatsoa uupuvan opintosuorituksen. Tämä sijaitsee detail-solussa (vasemmalla alhalla), jokaisen uupuvan opintosuorituksen yhteydessä.

3.23.1 Kuvaus

```
public class YlenkatsoJButton
extends javax.swing.JButton
```

3.23.2 Konstruktorien yhteenveto

YlenkatsoJButton(UupuvaOpintojakso) Luo tämän nappulan.

3.23.3 Konstruktorit

- **YlenkatsoJButton**

```
public YlenkatsoJButton (
fi.helsinki.cs.ilpo.UupuvaOpintojakso uupuva )
```

- **Kuvaus**

- Luo tämän nappulan.

- **Parametrit**

- * `uupuva` – UupuvaOpintoJakso, mitä tällä napilla ylenkatsotaan.

Chapter 4

Pakkaus fi.helsinki.cs.ilpo.db

<i>Pakkaus Sisältö</i>	<i>Sivu</i>
Luokat	
Parseri	52
Parsii kysymykset ja vastaukset.	
Tietokantakyselija	53
Tietokantakyselija sisältää SQL-kyselyt, joita Ilpo käyttää.	
Tietokantayhteys	58
Yhteys Ilmo/Desmond tietokantaan	

4.1 Luokka Parseri

Parsii kysymykset ja vastaukset.

4.1.1 Kuvaus

```
public class Parseri
extends java.lang.Object
```

4.1.2 Konstruktorien yhteenveto

Parseri(Tietokantakyselija, Kurssi) Luo parseri olion

4.1.3 Metodien yhteenveto

haeRadiobuttonKysymykset() Metodi hakee radiobutton tyypiset kysymykset

parsiiKysymykset() Metodi parsii kaikki kysymykset

parsiiVastaukset(Opiskelija, Kurssi) Metodi parsii kaikki vastaukset

4.1.4 Konstruktorit

- **Parseri**

```
public Parseri( Tietokantakyselyija tkk,
fi.helsinki.cs.ilpo.Kurssi k )
```

- **Kuvaus**

Luo parseri olion

- **Parametrit**

- * **tkk** – tietokantakyselyija
- * **k** – Kurssi olio

4.1.5 Metodit

- **haeRadiobuttonKysymykset**

```
public java.util.LinkedList haeRadiobuttonKysymykset (
)
```

- **Kuvaus**

Metodi hakee radiobutton tyyppiset kysymykset

- **Palauttaa** – LinkedList

- **parsiKysymykset**

```
public void parsiKysymykset ( )
```

- **Kuvaus**

Metodi parsii kaikki kysymykset

- **parsiVastaukset**

```
public void parsiVastaukset (
fi.helsinki.cs.ilpo.Opiskelija o,
fi.helsinki.cs.ilpo.Kurssi k )
```

- **Kuvaus**

Metodi parsii kaikki vastaukset

- **Parametrit**

- * **o** – Opiskelija olio
- * **k** – Kurssi olio

4.2 Luokka Tietokantakyselyija

Tietokantakyselyija sisältää SQL-kyselyt, joita Ilpo käyttää. Mikäli vastauksia ei tarvitse erikseen parsia, luo tietokantakyselyija vastaavat Ilpon oliot suoraan kyselyn vastauksessa. Parsimista vaativien kyselyjen vastaukset annetaan ResultSet-olioina parserille.

4.2.1 Kuvaus

```
public class Tietokantakyselija
  extends java.lang.Object
```

4.2.2 Konstruktorien yhteenveto

Tietokantakyselija(Kurssi, Tietokantayhteys) Luo Tietokantakyselija-olion.

4.2.3 Metodien yhteenveto

main(String[])

noudaKaikkiOpintojaksot() Noutaa kaikki Ilmo-kannasta löytyvät (tktl:n) opintojaksot ja luo niistä Opintojakso-oliot
SQL kysely:

```
SELECT kurssikoodi, nimi_suomi, opintoviikot
FROM opintojakso
```

noudaKayttajatunnukset() SQL-lause, joka noutaa kaikkien desmondista löytyvien opiskelijoiden opiskelijanumero-käyttäjätunnus parit.

noudaKurssinIlmoryhmat() Noutaa kaikki kurssiin liittyvät Ilmo-ryhmät sekä niihin liittyvät henkilökunnan jäsenet

noudaKurssinIlmoVastaukset() Noutaa kurssille ilmoittautuneiden Ilmoon kurssi-ilmottautumisen yhteydessä kirjoittamat vastaukset

noudaKurssinKysymykset() SQL-lause, joka noutaa kurssin Ilmo-kysymykset parseroitaviksi.

noudaKurssinOpiskelijat() Noutaa kaikkien kurssille ilmoittautuneiden opiskelijoiden henkilötiedot. Opiskelija-oliot luodaan Ilmo- ja Desmond kannoista löytyvien tietojen perusteella.

noudaOpiskelijanVastaukset(String) SQL-lause, joka noutaa opiskelijan Ilmo-vastaukset parseroitaviksi.

paivitaKurssinOpiskelijat() Päivittää urssille ilmoittautuneiden opiskelijoiden muuttuneet henkilötiedot ja merkitsee ne opiskelijat poistetuiksi, jotka ovat peruneet ilmonsä.

paivitaPaaainesuoritukset(Opiskelija, Hashtable) Linkittää Opintojakso-oliot Opiskelija-olioihin.

yhdistaPaaainesuorituksiin(Opiskelija, Hashtable) Linkittää Opintojakso-oliot Opiskelija-olioihin.

yhdistaSivuainesuorituksiin(Opiskelija) Noutaa opiskelijan suorittamat sivuaineopinnot, sekä luo niistä tarvittavat Sivuaine- ja SuoritettuSivuainekurssi -oliot, jotka liittävät Opiskelija-olioon.

4.2.4 Konstruktorit

- **Tietokantakyselija**

```
public Tietokantakyselija( fi.helsinki.cs.ilpo.Kurssi
k, Tietokantayhteys yhteys )
```

- **Kuvaus**

Luo Tietokantakyselija-olion.

- **Parametrit**

- * k – Kurssi-jota ryhmitellään
- * yhteys – Tietokantayhteys, jota kyselijä käyttää urkintoihinsa

4.2.5 Metodit

- **main**

```
public static void main( java.lang.String[] args )
heittää java.lang.Exception
```

- **noudaKaikkiOpintojaksot**

```
public java.util.Hashtable noudaKaikkiOpintojaksot( )
```

- **Kuvaus**

Noutaa kaikki Ilmo-kannasta löytyvät (tktl:n) opintojaksot ja luo niistä Opintojakso-oliot

SQL kysely:

```
SELECT kurssikoodi, nimi_suomi, opintoviikot FROM
opintojakso
```

- **Palauttaa** – Hajautustaulun, jossa on kaikki tietojenkäsittelytieteen opintojaksot. Avaimena toimii kurssikoodi.

- **noudaKayttajatunnukset**

```
public java.util.Hashtable noudaKayttajatunnukset( )
```

- **Kuvaus**

SQL-lause, joka noutaa kaikkien desmondista löytyvien opiskelijoiden opiskelijanumero-käyttäjätunnus parit.

- **Palauttaa** – Opiskelijanumero-käyttäjätunnus parit hajautustaulussa. Avain on opiskelijanumero.

- **noudaKurssinIlmoryhmat**

```
public java.util.Collection noudaKurssinIlmoryhmat( )
```

- **Kuvaus**

Noutaa kaikki kurssiin liittyvät Ilmo-ryhmät sekä niihin liittyvät henkilökunnan jäsenet

- **Palauttaa** – IlmoRyhma-oliot, joihin on liitetty vastaavat Henkilokuntalainen-oliot.

- **noudaKurssinIlmoVastaukset**

```
public java.sql.ResultSet noudaKurssinIlmoVastaukset( )
```

- **Kuvaus**

Noutaa kurssille ilmoittautuneiden Ilmoon kurssi-ilmottautumisen yhteydessä kirjoittamat vastaukset

- **Palauttaa** – ResultSet

- **noudaKurssinKysymykset**

```
public java.sql.ResultSet noudaKurssinKysymykset( )  
heittää java.lang.Exception
```

- **Kuvaus**

SQL-lause, joka noutaa kurssin Ilmo-kysymykset parseroitaviksi.

- **Palauttaa** – Ilmo-kysymykset ResultSet-oliossa.

- **noudaKurssinOpiskelijat**

```
public java.util.Collection noudaKurssinOpiskelijat( )
```

- **Kuvaus**

Noutaa kaikkien kurssille ilmoittautuneiden opiskelijoiden henkilötiedot.Opiskelija-oliot luodaan Ilmo- ja Desmond kannoista löytyvien tietojen perusteella. Metodilla ei suoriteta päivityksiä.
SQL kysely:

```
SELECT op.hetu, op.opnro, op.sukunimi ||' '  
||op.etunimi as nimi, op.sahkopostiosoite,  
op.aloitusvuosi, op.paa_aine, os.ryhma_nro FROM  
osallistuminen os, opiskelija op WHERE os.hetu =  
op.hetu AND os.voimassa = 'K' AND os.kurssikoodi =  
kk AND os.lukuvuosi = lv AND os.lukukausi=lk AND  
os.kurssi_nro = knro
```

- **Palauttaa** – Opiskelijat Collection-oliossa.

- **noudaOpiskelijanVastaukset**

```
public java.sql.ResultSet noudaOpiskelijanVastaukset(  
java.lang.String hetu )
```

- **Kuvaus**

SQL-lause, joka noutaa opiskelijan Ilmo-vastaukset parseroitaviksi.

- **Parametrit**

* hetu – Opiskelijan henkilötunnus

– **Palauttaa** – Ilmo-vastaukset ResultSet-oliassa.

- **paivitaKurssinOpiskelijat**

```
public void paivitaKurssinOpiskelijat ( )
```

- **Kuvaus**

Päivittää urssille ilmoittautuneiden opiskelijoiden muuttuneet henkilötiedot ja merkitsee ne opiskelijat poistetuiksi, jotka ovat peruneet ilmonsä. Tarvittavat uudet Opiskelija-oliot (jälki-ilmoittautuneet) luodaan Ilmo- ja Desmond kannoista löytyvien tietojen perusteella.

- **paivitaPaaainesuoritukset**

```
public void paivitaPaaainesuoritukset (
fi.helsinki.cs.ilpo.Opiskelija opiskelija,
java.util.Hashtable opintojaksot )
```

- **Kuvaus**

Linkittää Opintojakso-oliot Opiskelija-olioihin. Noutaa opiskelijan pääainesuoritukset, joista generoi SuoritettuOpintojakso-olioita Opintojakso-olioita hyväksikäyttäen.

- **Parametrit**

- * opiskelija – Opiskelija, jonka opintosuoritukset päivitetään.
- * opintojaksot – Kaikki kannasta löytyvät tk1:n opintojaksot

- **yhdistaPaaainesuorituksiin**

```
public void yhdistaPaaainesuorituksiin (
fi.helsinki.cs.ilpo.Opiskelija opiskelija,
java.util.Hashtable opintojaksot )
```

- **Kuvaus**

Linkittää Opintojakso-oliot Opiskelija-olioihin. Noutaa opiskelijan pääainesuoritukset, joista generoi SuoritettuOpintojakso-olioita Opintojakso-olioita hyväksikäyttäen.

- **Parametrit**

- * opiskelija – Opiskelija, jonka opintosuoritukset noukitaan
- * opintojaksot – Kaikki kannasta löytyvät tk1:n opintojaksot

- **yhdistaSivuainesuorituksiin**

```
public void yhdistaSivuainesuorituksiin (
fi.helsinki.cs.ilpo.Opiskelija opiskelija )
```

- **Kuvaus**

Noutaa opiskelijan suorittamat sivuaineopinnot, sekä luo niistä tarvittavat Sivuaine- ja SuoritettuSivuainekurssi -oliot, jotka liittää Opiskelija-olioon. Metodilla voidaan myös päivittää sivuaineopinnot.

– Parametrit

* opiskelija – Opiskelija, jonka suoritukset noudetaan.

4.3 Luokka Tietokantayhteys

Yhteys Ilmo/Desmond tietokantaan

4.3.1 Kuvaus

```
public class Tietokantayhteys
extends java.lang.Object
```

4.3.2 Konstruktorien yhteenveto

Tietokantayhteys(String, String, String, String, String, String, String) Luo uuden tietokantayhteyden.

4.3.3 Metodien yhteenveto

```
main(String[])
suljeYhteys() Sulkee yhteyden tietokantaan.
suoritaKysely(String) Suorittaa SQL-kyselyn.
```

4.3.4 Konstruktorit

- **Tietokantayhteys**

```
public Tietokantayhteys( java.lang.String
ajuriluokanNimi, java.lang.String ajurinNimi,
java.lang.String tietokannanOsoite, java.lang.String
tietokannanPortti, java.lang.String tietokannanNimi,
java.lang.String kayttaja, java.lang.String salasana )
heitteä java.lang.Exception
```

– **Kuvaus**

Luo uuden tietokantayhteyden. Jos yhteyden luominen ei suju (väärä salasana / kanta nurin) lentää poikkeus, jonka koppaaja kaataa Ilpon herjaten käyttäjää.

– **Parametrit**

- * ajuriluokanNimi – Luokan nimi, josta varsinainen ajuri ladataan
- * ajurinNimi – Käytettävän tietokanta-ajurin nimi
- * tietokannanOsoite – Tietokannan osoite
- * tietokannanPortti – Tietokannan portti

- * tietokannanNimi – Tietokannan nimi
- * kayttaja – Tietokannan käyttäjä
- * salasana – Tietokannan salasana

– **Throws**

- * `java.lang.Exception` – jos ajuriluokkaa ei löydy (`ClassNotFoundException`) tai tietokantayhteyden avaamisessa ongelmia (`SQLException`)

4.3.5 Metodit

- **main**

```
public static void main( java.lang.String[] args )
```

- **suljeYhteys**

```
public boolean suljeYhteys( )
```

- **Kuvaus**

Sulkee yhteyden tietokantaan.

- **Palauttaa** – boolean tieto onko yhteys suljettu.

- **suoritaKysely**

```
public java.sql.ResultSet suoritaKysely(
java.lang.String kysely ) heittää
java.sql.SQLException
```

- **Kuvaus**

Suorittaa SQL-kyselyn. Jos tietokantayhteys on katkennut niin se avataan automaattisesti uudelleen ennen kuin kysely yritetään suorittaa.

- **Parametrit**

- * `kysely` – Suoritettava SQL-kysely merkkijonona.

- **Palauttaa** – Kyselyn tulos `ResultSet`-oliossa.

- **Throws**

- * `java.sql.SQLException` – jos yhteydessä tai SQL-lauseessa vikaa

Chapter 5

Pakkaus fi.helsinki.cs.ilpo.io

<i>Pakkaus Sisältö</i>	<i>Sivu</i>
Luokat	
HTMLKirjoitin	60
Kirjastoluokka tehdyn ryhmäjaon saamiseksi ulos HTML-formaatissa	
KeskeneraistenRyhmiinKasittelija	61
Kirjastoluokka keskeneräisten ryhmittelyjen tallentamiseen ja lataamiseen.	
OpiskelijanumeroTulostin	63
Kirjastoluokka, jonka ainoalla metodilla tulostetaan tekstitiedostoon ryhmiteltävien opiskelijoiden opiskelijanumerot.	
Tiedostokirjoittaja	64
Luokka, jonka avulla hoidetaan tiedostojen varsinainen kirjoitus.	

5.1 Luokka HTMLKirjoitin

Kirjastoluokka tehdyn ryhmäjaon saamiseksi ulos HTML-formaatissa

5.1.1 Kuvaus

```
public final class HTMLKirjoitin
extends java.lang.Object
```

5.1.2 Metodien yhteenveto

kirjoitaHTML(Kurssi, String) Kirjoittaa ryhmäjaon HTML-sivun tyngäksi, eli pelkän table-elementin sisällön.
main(String[])

5.1.3 Metodit

- **kirjoitaHTML**

```
public static boolean kirjoitaHTML (
    fi.helsinki.cs.ilpo.Kurssi kurssi, java.lang.String
    tiedostonNimi )
```

- **Kuvaus**

Kirjoittaa ryhmäjaon HTML-sivun tyngäksi, eli pelkän table-elementin sisällön.

- **Parametrit**

- * kurssi – Ryhmiteltävä kurssi-olio.
- * tiedostonNimi – Kirjoitettavan tiedoston nimi

- **Palauttaa** – Ilmaisee, josko operaatio sujui onnellisten tähtien alla.

- **main**

```
public static void main( java.lang.String[] args )
```

5.2 Luokka KeskenäraistenRyhmiinKasittelija

Kirjastoluokka keskeneräisten ryhmittelyjen tallentamiseen ja lataamiseen. Varsinainen kirjoitus tehdään Tiedostokirjoittaja-kirjastoluokan avulla.

5.2.1 Kuvaus

```
public class KeskenäraistenRyhmiinKasittelija
    extends java.lang.Object
```

5.2.2 Konstruktorien yhteenveto

KeskenäraistenRyhmiinKasittelija(String)

5.2.3 Metodien yhteenveto

annaTiedostonNimi() Get the value of tiedostonNimi

asetataTiedostonNimi(String) Set the value of tiedostonNimi.

lataaRyhmittely() Lataa ryhmittelyn levyältä.

main(String[])

tallennaRyhmittely(Kurssi) Tallentaa ryhmittelyn asetettuun tiedostoon.

5.2.4 Konstruktorit

- **KeskeneraistenRyhmienKasittelija**

```
public KeskeneraistenRyhmienKasittelija( java.lang.String
    tiedostonNimi )
```

5.2.5 Metodit

- **annaTiedostonNimi**

```
public java.lang.String annaTiedostonNimi( )
```

- **Kuvaus**
Get the value of tiedostonNimi
- **Palauttaa** – the value of tiedostonNimi

- **asetaNimi**

```
public void asetaNimi( java.lang.String value
    )
```

- **Kuvaus**
Set the value of tiedostonNimi.
- **Parametrit**
* value – Tiedoston nimi, jota halutaan käsitellä.

- **lataaRyhmittely**

```
public fi.helsinki.cs.ilpo.Kurssi lataaRyhmittely( )
    heittää java.lang.Exception
```

- **Kuvaus**
Lataa ryhmittelyn levyltä.
- **Palauttaa** – Viite ladattuun Kurssi-olioon.
- **Throws**
* java.lang.Exception – Jos tiedoston latauksessa tapahtuu virhe

- **main**

```
public static void main( java.lang.String[] args )
```

- **tallennaRyhmittely**

```
public void tallennaRyhmittely(
    fi.helsinki.cs.ilpo.Kurssi kurssi ) heittää
    java.lang.Exception
```

- **Kuvaus**
Tallentaa ryhmittelyn asetettuun tiedostoon.

- **Throws**

- * `java.lang.Exception` – Jos tiedoston tallentamisessa tapahtuu virhe

5.3 Luokka OpiskelijanumeroTulostin

Kirjastoluokka, jonka ainoalla metodilla tulostetaan tekstitiedostoon ryhmiteltävien opiskelijoiden opiskelijanumerot. Varsinainen kirjoitus tehdään Tiedostokirjoittaja-kirjastoluokan avulla.

5.3.1 Kuvaus

```
public final class OpiskelijanumeroTulostin
extends java.lang.Object
```

5.3.2 Metodien yhteenveto

kirjoitaOpiskelijat(Collection, String) Tulostaa Opiskelija-olioiden opiskelijanumerot tekstitiedostoon 1/rivi.
main(String[])

5.3.3 Metodit

- **kirjoitaOpiskelijat**

```
public static boolean kirjoitaOpiskelijat (
java.util.Collection opiskelijat, java.lang.String
tiedostonNimi )
```

- **Kuvaus**

- Tulostaa Opiskelija-olioiden opiskelijanumerot tekstitiedostoon 1/rivi.

- **Parametrit**

- * `opiskelijat` – Opiskelija-oliot Collection-rajapinnan toteuttavassa joukossa, joiden opiskelijanumerot halutaan kirjoittaa

- * `tiedostonNimi` – Tiedosto, johon kirjoitetaan

- **Palauttaa** – Ilmoittaa onnistuiko operaatio

- **main**

```
public static void main( java.lang.String[] args )
```

5.4 Luokka Tiedostokirjoittaja

Luokka, jonka avulla hoidetaan tiedostojen varsinainen kirjoitus. Eli muut luokat käyttävät tätä avukseen.

5.4.1 Kuvaus

```
public class Tiedostokirjoittaja
extends java.lang.Object
```

5.4.2 Konstruktorien yhteenveto

Tiedostokirjoittaja(String) Luo uuden Tiedostokirjoittaja-olion.

5.4.3 Metodien yhteenveto

avaaTiedosto() Avaa tiedoston käsittelyä varten
kirjoita(String) Kirjoittaa tiedostoon merkkijonon
onkoOlemassa() Tarkistaa onko tiedosto olemassa
suljeTiedosto() Sulkee avoimena olevan tiedoston

5.4.4 Konstruktorit

- **Tiedostokirjoittaja**

```
public Tiedostokirjoittaja ( java.lang.String tiedostonNimi
)
```

 - **Kuvaus**
Luo uuden Tiedostokirjoittaja-olion.
 - **Parametrit**
 - * tiedostonNimi – Avattavan tiedoston nimi

5.4.5 Metodit

- **avaaTiedosto**

```
public boolean avaaTiedosto ( )
```

 - **Kuvaus**
Avaa tiedoston käsittelyä varten
 - **Palauttaa** – True, mikäli avaus onnistui
- **kirjoita**

```
public boolean kirjoita ( java.lang.String merkkijono )
```

- **Kuvaus**
Kirjoittaa tiedostoon merkkijonon
- **Parametrit**
 - * merkkijono – kirjoitettava tieto
- **Palauttaa** – true, mikäli kirjoitus onnistui.

- **onkoOlemassa**

```
public boolean onkoOlemassa ( )
```

- **Kuvaus**
Tarkistaa onko tiedosto olemassa
- **Palauttaa** – True, mikäli Tiedosto löytyi.

- **suljeTiedosto**

```
public void suljeTiedosto ( )
```

- **Kuvaus**
Sulkee avoinna olevan tiedoston