



# Kurssitulokset

## *Kurkidokumentaatio*

### **Ohjelmistotuotantoprojekti - kesä 2012**

Tietojenkäsittelytieteen laitos

Helsingin yliopisto

Kasper Hirvikoski

Kenny Heinonen

Arvi Nikunlassi

Hannu Päiveröinen

Natalia Tuulos

# Aiheen kuvaus

Kurssitulosten ja suorituspalautteen sähköistäminen

[http://www.cs.helsinki.fi/group/ohu/ke-2012/aihe\\_kurssitulokset.html](http://www.cs.helsinki.fi/group/ohu/ke-2012/aihe_kurssitulokset.html)

## Sähköpostin lähetys Kurjesta

### Tehdyt muutokset:

**checklist99\_1.vm** (suomenkielinen viesti)  
**checklist99\_2.vm** (englanninkielinen viesti)  
**checklist99\_3.vm** (ruotsinkielinen viesti)

#### Freeze.java

```
String arvosana = ""; //opiskelijan arvosana
Vector <Student> students = course.getStudents(); //opiskelijan tiedot
String studentToAddress = ""; //opiskelijan sähköpostiosoite
String msgStudent = ""; //viesti suomeksi
String msgStudentEngl = ""; //viesti englanniksi
String msgStudentRuotsi = ""; //viesti ruotsiksi
String subjectStudent = subject; //sähköpostiviestin aihe

StringWriter suomi = new StringWriter();
StringWriter englanti = new StringWriter();
StringWriter ruotsi = new StringWriter();
Velocity.mergeTemplate("checklist99_1.vm", results, suomi);
msgStudent = suomi.toString();
Velocity.mergeTemplate("checklist99_2.vm", results, englanti);
msgStudentEngl = englanti.toString();
Velocity.mergeTemplate("checklist99_3.vm", results, ruotsi);
msgStudentRuotsi = ruotsi.toString();

//*****Ilmoitus tuloksista kurssille osallistuneille Opiskelijoille*****
MimeMessage messageSt = null;

for(int i = 0; i < students.size(); i++) {

    studentToAddress = students.get(i).getEmail();
    arvosana = students.get(i).getGrade();

    if (arvosana != null && studentToAddress != null){
        messageSt = new MimeMessage(s);
        messageSt.setFrom(from);
        messageSt.setSubject(subjectStudent);
        //suomenkielinen viesti
        messageSt.setText("Sinulle on annettu seuraava arvosana: "+arvosana + msgStudent);
        //suomenkielinen + englanninkielinen viesti
        messageSt.setText("Sinulle on annettu seuraava arvosana: "+arvosana + msgStudent+
            "You have been given the following grade: " + arvosana + msgStudentEngl);
        //ruotsinkielinen viesti
```

```

        messageSt.setText("Du har fått följande betyg: "+arvosana +msgStudentRuotsi);

        messageSt.addRecipient(Message.RecipientType.TO, new InternetAddress( studentToAddress )
    );
        Transport.send(messageSt);
    }
}

```

#### web.xml

```

<context-param>
<param-name>configurationFile</param-name>
<param-value>Oma_polku/kurki/src/main/webapp/WEB-INF/kurki.cnf</param-value>
</context-param>

```

## Kurki

### Course.java

```

// Analysoi kurssin perustiedot.
public Course( CourseInfo courseInfo ) throws SQLException, NullPointerException, ClassNotFoundException
{

    if ( courseInfo == null ) throw new NullPointerException();

    this.courseInfo = courseInfo;

    load();
}
protected Vector students = null;
protected CourseInfo courseInfo;

public synchronized CourseInfo getCourseInfo() {return this.courseInfo;}

public void newSearch() {
    newSearch( false );
}

public synchronized void newSearch( boolean force ) {

    if ( force ) { this.students = null; }
    this.orderBy = ORDER_BY_NAME;
    this.group.clear(); //protected LinkedList groups = new LinkedList();
    this.lname.clear(); //protected LinkedList lname = new LinkedList();
    this.ssn.clear(); //protected LinkedList ssn = new LinkedList();
    this.lnameFrom = this.lnameTo = null;
}

```

### CourseInfo

```

public String getName() { return this.desc; }

```

### ComparableOptionAdapter

```

protected String desc;

```

**Student**

```
this.desc = this.lname+" ", " +this.fname.substring(0, i );
```

**Kurki**

**public class Teacher** (opettajan tiedot)

**Atribuutit**

- **String:** htunnus, etunimet, sukunimi, hetu, ktunnus;

**Konstruktori:**

- public Teacher ( String htunnus, String etunimet, String sukunimi, String hetu, String ktunnus )

**public class Student** extends ComparableOptionAdapter implements Serializable  
(opiskelijan tiedot)

**Atribuutit**

- **String:** address, bday, email, fname, lname, major, phone, prevlname, state, sno, ssn, language
- **int:** group, groupId, startYear, xtrTotal, creditsNewUnits
- **int[]:** scoreTotal, validScoreCount, xtrScore

**Konstruktorit:**

- public Student( int id, String ssn )
- public Student( int id, String ssn, String name )
- public Student( int id, String ssn, String lname, String fname, int group )
- public Student( int id, String ssn, String lname, String fname, String email, int group )

**Getterit:** getAddress, getBirthday, getCreditsNew, getGroup, getGroupId, getGrade, getId, getEmail, getEnrolment, getFName, getLName, getLanguage, getMajor, getName (sukunimi+etunimi), getPhone, getPrevLName, getScore, getScores, getScoreTotal, getScoreUpdate, getSNO, getSSN, getSSNID, getSubSSN, getStartYear, getXtrScore, getXtrTotal.

**Setterit:** setAddress, setCreditsNew, setCreditsOld, setEnrolment, setGroup, setGroupId, setHasScores, setMajor, setLanguage, setPhone, setPrevLName, setSNO, setScore, setStartYear, setXtrTotal, setXtrScore.

**Jäädytykseen liittyvät metodit:**

```

public boolean defreeze() {
    if ( this.isFrozen() ) { this.state = "S"; return true; }
    else return false;
}

public void freeze() { this.state = "J"; }

public boolean isDefrost() { return this.state != null && this.state.equals("S"); }

public boolean isFrozen() { return this.state != null && this.state.equals("J"); }

```

**public class CourseInfo** extends ComparableOptionAdapter (kurssin tiedot)

#### Atribuutit

- **int**: userRole, year, cno
- **String**: ccode, term, type
- **Boolean**: isFrozen

#### Konstruktorit:

- public CourseInfo( String id, String name )
- public CourseInfo( String id, String name, int userRole )
- public CourseInfo( String ccode, int year, String term, String type, int cno, String name, int userRole )

#### Jäädätykseen liittyvät metodit:

- public void freeze() { this.isFrozen = true; }
- public boolean isFrozen() { return this.isFrozen; }

**Getterit:** getId, getName, getYear, getCNO, getCCode, getTerm, getType, getValidServices, getUserRole.

**public class Session** implements java.io.Serializable (istunto)

- Kuinka pitkään kurssin tietoihin saa tehdä muutoksia.
- Toteutettujen toimintojen perustietojen määrittely <toiminnon id>, <kuka saa suorittaa>, <kuvaus>
- Super users
- Käsittelyssä oleva kurssi
- Kurssit (CourseInfo), joita käyttäjä on oikeutettu päivittämään
- Valittu palvelu
- Estetään ilmentymien luonti ulkopuolisilta
- Onko käyttäjä jo valinnut käsiteltävän kurssin
- Suorittaa tarvittavat alustukset ja palauttaa uuden ilmentymän tästä luokasta
- Kurssin <tt>cid</tt> perustiedot
- Käsiteltävissä olevan kurssin perustiedot

- Kaikkien niiden kurssien perustiedot, joilla käyttäjä ohjaajana
- Käyttäjän päivitysoikeudet
- Selvitetään, millä kurssilla käyttäjä on luennoitsijana, laskariohjaajana tai käytännön vastuushenkilönä sekä mitä päivityksiä kyseinen käyttäjä saa kurssille tehdä.
- Onko käyttäjä oikeutettu suorittamaan `<tt>action</tt>`-päivityksen
- Valitaan kurssi `<tt>cinfo</tt>` käsittelyyn
- Valitaan kurssi `<tt>cid</tt>` käsittelyyn

**public class Course** implements Option, Serializable (kurssi)

- UPDATE osallistuminen
- UPDATE kurssi
- Opiskelijoiden haku ryhmän perusteella
- Opiskelijoiden haku sukunimen perusteella
- Opiskelijoiden haku opiskelijanumeron perusteella
- Opiskelijoiden haku henkilötunnuksen perusteella
- Opiskelijoiden haku sukunimen perusteella annetulta väliltä
- Laskarit
- Harjoitustyöt
- Kokeet
- Arvosana
- Opintopisteet

```
+COURSE_WHERE_SQL
+"      AND voimassa = 'K'\n"
+"      AND (jaassa != 'J' OR jaassa IS null)\n"
+"      AND hetu = ?",
protected String scale = "K"; // K = 1-5, E = hyväksytty/hylätty
```

## Muut luokat:

**public class DBConnectionManager** implements Serializable (tietokantayhteyden avaaminen ja sulkeminen)

**public class Configuration** (konfigurointi: getProperty, setProperty, loadProperties)

**public class MultiValueCounter** (laskuri)

ComparableOptionAdapter-luokan aliluokat:

**public class Convention** extends ComparableOptionAdapter (course-luokassa sovitut säännöt)

**public class Score** extends ComparableOptionAdapter implements Serializable (pisteet)

**public class Part** extends ComparableOptionAdapter implements Serializable (osat)

**public class Offering** extends ComparableOptionAdapter implements Serializable (osasuoritukset)

## Poikkeusluokat:

**public class InitFailedException** extends Exception

```
public class InvalidScoreException extends Exception
public class NullParameterException extends Exception
public class CourseNotDefinedException extends Exception
```

### Rajapintaluokka:

```
public interface Log
```

## kurki.servlet

**public abstract class AbstractVelocityServiceProvider** extends ServiceAdapter (Abstrakti Velocity palveluntarjoaja)

Käsittelijäluokka (**AbstractVelocityServiceProvider**) määrittelee suorituspyyntöjen rajapinnan. Aliluokissa käsitellään handleRequest()-metodeissa ne suorituspyynnöt, joista ollaan vastuussa tai välitetään suorituspyyntö seuraajalle.

- Käsittelee http-pyynnön kutsumalla handleService-metodia ja huolehtimalla Templaten käsittelyyn liittyvistä virhetilanteista.

AbstractVelocityServiceProvider aliluokat:

**public class Checklist** extends AbstractVelocityServiceProvider

**public class CourseBasics** extends AbstractVelocityServiceProvider implements Serializable (Kurssin perustiedot)

**public class Entry** extends AbstractVelocityServiceProvider (merkintöjen tekeminen)

- Opiskelijasuodatin
- Http-parametrit
- Kirjattavan osasuoritteen valinta
- Opiskelijat: ryhmän, henkilötunnuksen tai sukunimen mukaan
  - Haku ryhmän mukaan, jos annettu kokonaisluku, muuten haetaan sukunimen tai henkilötunnuksen mukaan
- Vaihdetään kirjattavan osasuoritteen tyyppiä
- Kirjattavan osasuoritteen perustietojen määrittäminen
- Kirjataan annetut pisteet
- Opiskelijan sulatus

**! public class Freeze** extends AbstractVelocityServiceProvider

- Tuloslista jäädytyksen tekijälle sekä Oodi-siirroista vastaavalle
  - Listaan mukaan otettavat tiedot:
    - Opiskelijanro
    - Nimi

- Laskaripisteet
- Laskaripisteiden sarakeotsake
- Harjoitustyöpisteet
- Harjoitustyöpisteiden sarakeotsake
- Koepisteet
- Koepisteiden sarakeotsake
- Yhteispisteet
- Yhteispisteiden sarakeotsake
- Hyväksyjän allekirjoitus
- Tilastot
- Arvosana
- Arvosanan sarakeotsake
- Hyväksytyt
- Hylätyt
- Korjauslista (vain muutokset)
- Jos jäädytys onnistuu, lähetetään sähköpostia  

```
if ( course.freeze( 0 ) ) {.....}
```

  - Jäädytysilmoitus Oodisiirroista vastaavalle ja jäädytyksen suorittajalle
  - Jos halutaan korjata opiskelijan puutteellisia tai virheellisiä pisteitä, sen voi tehdä sulattamalla ensin kyseisen opiskelijan. Tämän jälkeen kurssi on kuitenkin jäädytettävä uudelleen.

**public class Grades** extends AbstractVelocityServiceProvider implements Serializable  
(arvosanoja)

- Tallentaa lomakkeen tiedot
  - Suorituspäivä
  - Osasuoritusten lukumäärä
  - Osuuden hyväksymisraja
  - Pisterajojen muokkauspyyntö
  - Pisterajojen tallennus
  - Arvostelu

**public class Participants** extends AbstractVelocityServiceProvider implements Serializable  
(osallistujat)

- Opiskelijan hakeminen
- Opiskelijan tietojen muokkauspyyntö
- Opiskelijan poisto kurssilta
- Opiskelijan ryhmän muuttaminen
- Opiskelijan lisääminen kursille
- Opiskelijan sulatus (Jotta tehdyt muutokset kirjautuisivat myös Oodiin täytyy kurssi jäädyttää uudelleen)

**public class ResultList** extends AbstractVelocityServiceProvider

**VelocityServlet aliluokat:**



**public class Index** extends VelocityServlet implements Log, Serializable

- Konfiguraatio
- Lisätään tieto lokista konfiguraatioon
- SUPER USERS
- Palveluiden käsittelijöiden määrittely
- Istunnon hallinta
  - Aikaleiman tarkistus
  - Käyttäjän oikeudet
  - Automaattinen välitallennus - päällä vai poissa?

**public class ListMaker** extends VelocityServlet implements Log, Serializable

**public class LoginManager** extends VelocityServlet (kirjautumisen hallinta)

Muut:

**public class Logout** extends HttpServlet

**public class ServletMonitor** (istunnon synkronointi)

## **Service**

**public abstract class ComparableOptionAdapter** implements ComparableOption

ComparableOptionAdapter aliluokka:

**public class ServiceAdapter** extends ComparableOptionAdapter implements Service

## **Poikkeusluokat:**

**public class NullIdException** extends Exception

**public class ServiceAlreadyDefinedException** extends Exception

**public class ServiceNotDefinedException** extends Exception

**public class ServicesAlreadyDefinedException** extends Exception

**public class ServicesLockedException** extends Exception

**public class ServicesNotLockedException** extends Exception

**public class ServicesNotDefinedException** extends Exception

**public class UndefinedRoleException** extends Exception

## Rajapintaluokat:

**public interface** Option

**public interface Service** extends ComparableOption (palveluen salliminen roolien mukaan)

**public interface ComparableOption** extends Comparable, Option

**public class ServiceManager** (palvelujen hallinta)

- Luo identifioimattoman toiminnon, jonka avulla voidaan esim. hakea tunnettu (ja identifioitu) toiminto
- Määrittelee tunnetun toiminnon, joka on identifioitu tietylle oikeusluokalle (käyttäjärühmälle) ja tätä ylemmille
- Tietyn käyttäjärühmän päivitysoikeudet
- Palvelujen saanti roolien mukaan

## Luokkakaavio

