

# Nuottiarkisto

## Toteutusdokumentti

Ahti Syreeni  
ahti.syreeni@helsinki.fi  
Tietokantasovellusten harjoitustyö  
Ohjaaja: Samppa Saarela

HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteen laitos  
Helsinki 4.5.2001

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Ohjelmiston yleisrakenne</b>	<b>3</b>
2.1	Järjestelmän komponentit . . . . .	4
2.1.1	Etusivu . . . . .	4
2.1.2	Haku . . . . .	8
2.1.3	Nuotin lisäys arkistoon . . . . .	11
2.1.4	Nuotin poisto . . . . .	14
2.1.5	Rekisteröityminen ja henkilötietojen muuttaminen . . . . .	17
2.1.6	Apuluokat . . . . .	20
<b>3</b>	<b>Käyttöohje</b>	<b>24</b>
3.1	Ohjelman asennus . . . . .	24
3.2	Ohjelman käynnistys ja käyttö . . . . .	26
3.2.1	Nuottijulkaisujen haku arkistosta . . . . .	26
3.2.2	Hakutulosten katselu . . . . .	29
3.2.3	Rekisteröityminen . . . . .	30
3.2.4	Sisäänkirjoittautuminen . . . . .	30
3.2.5	Henkilötietojen muuttaminen . . . . .	31
3.2.6	Henkilötietojen poistaminen nuottiarkistosta . . . . .	31
3.2.7	Nuotin lisäys arkistoon . . . . .	32
3.2.8	Nuottijulkaisun poisto arkistosta . . . . .	33
3.2.9	Lopetus . . . . .	33
3.3	Arkiston hoitajalle . . . . .	33
3.3.1	Nuottijulkaisujen poisto . . . . .	34
<b>A</b>	<b>Tehtäväkuvaus</b>	<b>1</b>
<b>B</b>	<b>Lähdekoodit</b>	<b>2</b>
<b>C</b>	<b>Tietokannan asennuslauseet</b>	<b>54</b>

D Kuvat html-sivuista	55
E Suunnitteludokumentti	59

# 1 Johdanto

Nuottiarkisto on www-pohjainen nuottijulkaisujen tietojen säilömispaikka. Arkistoon kerätään Internetissä saatavilla olevien nuottijulkaisujen tietoja ja eri käyttäjät lisäävät tiedot itse. Nuottiarkistosta voidaan hakea nuottijulkaisuja erilaisin hakuehdoin ja nuottijulkaisuja voidaan sekä lisätä että poistaa. Tekijänoikeuskysymysten varmistamiseksi tallennetaan nuottiarkistoon myös nuottijulkaisun lisääjän tiedot. Itse nuottijulkaisuista nuottiarkistoon talletetaan vain URL-osoite, jossa nuottijulkaisu on saatavilla.

Järjestelmän on tarkoitus palvella muusikkoja, säveltäjiä sekä musiikin harrastajia. Palvelun avulla muusikot ja musiikin harrastajat löytävät helposti uutta ilmaista soitettavaa Internetistä. Toisaalta säveltäjät saavat teoksensa keskitetysti esille.

Järjestelmän ensisijainen tavoite on nopeuttaa nuottijulkaisujen etsimistä Internetistä. Samalla palvelun käyttäjät näkevät uusia kiinnostavia kappaleita ja toivon mukaan jatkuvasti laajentavat esiintymisohjelmistoaan. Järjestelmän tavoitteena on myös luoda uusi kanava musiikin julkaisemiseen Internetissä.

Järjestelmän toimii www-palvelimella käyttäen relaatiotietokantaa. Toteutuskielenä käytetään Java Servlet-teknologiaa. Käyttöliittymä on www-pohjainen ja se mahdollistaa pääsyn järjestelmään mistä vain, missä on käytössä www-selain. Järjestelmän toimii pienillä muutoksilla yleisimmissä relaatiotietokannoissa (Oracle, MS SQL-server.) ja yleisimmissä Java-servlettejä tukevissa palvelimissa ja selaimissa (MS Explorer, Mozilla ja Netscapen uusimmat versiot). Toteutuksessa on käytetty tyylimäärä-rittelyjä, joita tällä hetkellä täydellisesti tukevat ainoastaan MS Internet Explorerin uudemmat versiot. Sivujen ulkonäkö on siis optimoitu ainoastaan kyseiselle selaimelle, muissa selaimissa sivujen ulkonäkö voi vaihdella. Tämä ei kuitenkaan tarkoita sitä, että sovellusta ei voisi käyttää muilla selaimilla, mutta tällöin tulee lukea sivujen ohjetekstit huolella sillä joissakin selaimissa jotkin kenttien nimet voivat jäädä piiloon. Sovelmaa on testattu sekä MS Explorerilla että Netscape Communicator-selaimella.

On myös kokeiltu, että sovellusta voidaan välttävästi käyttää joillakin tekstipohjaisilla selaimilla (Lynx).

Toteutusvaiheessa ei ole tehty alkuperäisestä suunnitteludokumentista poikkeavia rajoituksia työn laajuuden suhteen. Toteutus siis kattaa kaikki suunnitteludokumentin käyttötapaukset. Teoksiin ei kuitenkaan ole lisätty laskureita, koska niitä ei suunnitteludokumentissa ole määritelty. Lisäksi monien käyttötapauksien monipuolinen toteutus on luonut enemmän toiminnallisuutta ja mahdollisuuksia kuin suunnitteluvaiheessa on esitetty. Tulee kuitenkin huomauttaa, että sovelluksessa suunnittelun mukaisesti ei ole mahdollista muuttaa teostietoja niiden tallettamisen jälkeen. Arkistonhoitajan oikeudet voidaan antaa vain suoraan tietokantaan kohdistuvilla SQL-kyselyillä.

Tämä ohjelma on toteutettu käyttäen Helsingin yliopiston Oracle 8.0-tietokantaa sekä Java Servlettejä tukevaa Helsingin yliopiston tietojenkäsittelytieteen laitoksen palvelinta,  
db.cs.helsinki.fi.

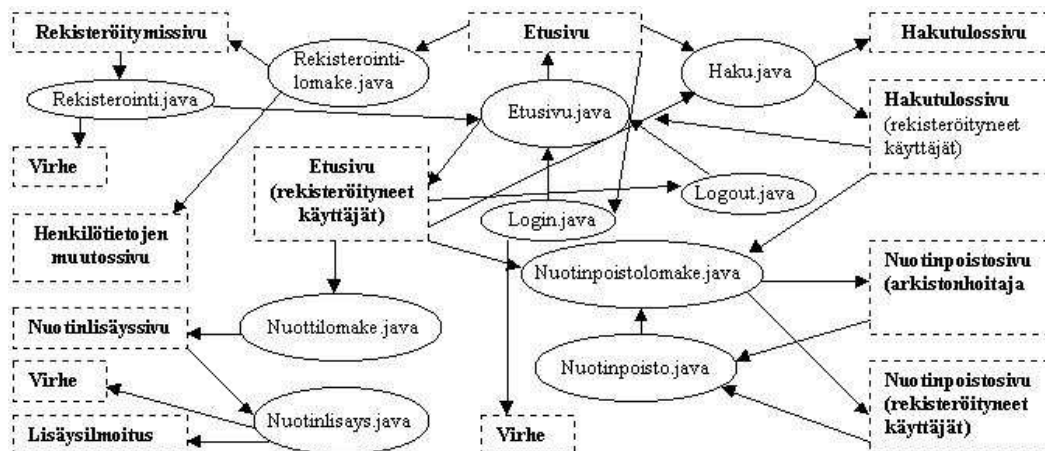
## 2 Ohjelmiston yleisrakenne

Ohjelma on toteutettu servlet-luokkien ja apukirjastoluokkien avulla. Jokaiselle toimenpiteelle on oma servletti (taulukko 1), joka generoi uuden sivun. Servletti generoi sivun kutsumalla sivuun liittyvien kirjastoluokkien metodeita, jotka palauttavat valmiita html-koodin pätkiä (taulukko 2). Tämä tarkoittaa sitä, että sivujen ulkoasu on helposti muutettavissa muuttamalla apuluokkaa, jossa palautettavat koodin pätkät ovat suurimmaksi osaksi html-koodina. Joissakin html-muotoiluissa apuluokat käyttävät myös erityisen Html-luokan palveluja. Sivun generoimiseen tarkoitettut apuluokat on nimetty siten, että niiden nimen alkuosa viittaa generoitavaan sivuun ja loppuosa kertoo, että luokka sisältää tekstiä. Siis esimerkiksi Etusivutekstit.java sisältää etusivun generoimiseen tarvittavat tekstit. On kuitenkin huomattava, että sama servletti generoi eri näköisen sivun rekisteröityneille käyttäjille ja näin ollen useimmat sivun generoinnin apuluokat sisältävät sekä rekisteröityneille että rekisteröitymättömille tarkoitettut html-tekstit. Sivujen apuluokissa on luokkavakioina määritelty kuvat, joita käytetään sivun generoimisessa. Nämä hakemistopolut on muutettava, jos hakemistorakennetta muutetaan tai kuvien talletuspaikka muuttuu.

Kuvaan 1 on hahmoteltu ohjelman yleinen rakenne. Siitä kuitenkin puuttuvat apuluokat, joista ei generoida omaa sivua. Tällaisia ovat jo mainittu Html-luokka, Tietokantaluokka, jossa määritetään käytettävä tietokanta, sekä Virheilmoitukset-luokka, joka palauttaa kaikki tarvittavat virheilmoitukset. Sovelluksessa käytetään myös taustakuvaa (tausta.jpg) sekä nuottiarkiston otsikkokuvaa (nuottiarkisto.jpg). Kuvasta 1 puuttuvat myös yhteyksien selitteet. Tämä siksi, että kuva säilyisi lukukelpoisena. On kuitenkin itsestään selvää, että rekisteröityneen käyttäjän jokaisella sivulla on samat linkit kaikkiin toimintoihin ja rekisteröitymättömällä käyttäjällä ei hakutulossivulla ole linkki kuin etusivulle.

Tarkemmat komponenttien kuvaukset on esitetty tuonnempana.

Tähdennettäköön vielä, että sekä Tietokanta-luokan tiedot että kaikkien htm-teksti-



Kuva 1: Nuottiarkiston komponenttirakenne

luokkien tiedot on päivitettävä siirrettäessä ohjelma uuteen ympäristöön.

## 2.1 Järjestelmän komponentit

Tässä luvussa on esitetty luokkakuvaukset järjestelmän komponenteille. Kuvaukset on järjestetty luotavien sivujen ja niiden toiminnallisuuden mukaan. Kaikki parameterit ovat String-muotoisia, jos toisin ei ole mainittu.

### 2.1.1 Etusivu

Jos käyttäjä ei ole rekisteröitynyt, etusivulla on lyhyt yleisesittely, hakupalvelun lomake, sisäänkirjoittautumislomake sekä ohjeet nuottijulkaisujen hakemiseen arkistosta. Sivulta on myös linkki rekisteröitymislomakkeeseen, joka käynnistää Rekisteröintilomake-servletin. Hakulomakkeen lähettäminen käynnistää Haku-servletin ja sisäänkirjoittautumislomakkeen lähettäminen käynnistää Login-servletin.

Jos käyttäjä on rekisteröitynyt, etusivu generoituu siten, että sivulla on yleisesittelyn sijasta ilmoitus siitä, että käyttäjä on sisäänkirjoittautunut sekä linkit rekisteröityneiden käyttäjien toimintoihin. Nuotinisäys-linkki käynnistää Nuottilomake-servletin, Nuotinpoisto-linkki Nuotinpoistolomake-servletin, Henkilötietojen muutta-

Etusivu.java (g,t)
Haku.java (g,t)
Login.java (t)
Logout.java
Nuotinisays.java (g,t)
Nuotinhoisto.java (t)
Nuotinhoistolomake.java (g,t)
Nuottilomake.java (g,t)
Rekisterointi.java (g,t)
Rekisterointilomake.java (g,t)

Taulukko 1: Nuottiarkiston servlet-luokat: g=generoi sivun, t=yhteydessä tietokantaan

Etusivutekstit.java *
Hakutulostekstit.java *
Html.java *
Nuotinhoistotekstit.java *
Nuottilomaketekstit.java *
Rekisterointitekstit.java *
Tietokanta.java
Virheilmoitukset.java *

Taulukko 2: Nuottiarkiston apuluokat (html-tekstiluokat merkitty tähdellä)



minen Rekisterointilomake-servletin ja Lopetus-linkki Logout-servletin. Rekisteröityneiden käyttäjien etusivulla ei ole sisäänkirjoittautumislomaketta. Sen sijaan hakuohjeet löytyvät sivun alalaidasta kuten rekisteröitymättömienkin etusivulla.

Seuraavassa kuvataan etusivuun läheisesti liittyvät java-luokat.

- Etusivu.java

Servletti (liite B) generoi etusivun sekä rekisteröityneille että rekisteröimättömille käyttäjille.

Servletti voidaan käynnistää URL-osoitteesta, nuotinlisäys- ja poistosivuilta, hakutulossivulta, sekä rekisterointilomake- ja henkilötietomuutossivuilta.

Servletti ei saa parametreja.

Servletti tarkistaa evästeen (Cookie) avulla, onko käyttäjä rekisteröitynyt. Hakulomaketta varten haetaan tietokannasta kaikki sillä hetkellä valittavissa olevat aihepiirit ja näiden tietojen perusteella valitaan Etusivutekstit-luokasta sopivat osat joko rekisteröityneiden tai rekisteröityneiden käyttäjien etusivulle. Jos tietokantayhteys on poikki, annetaan Virheilmoitukset-luokan palvelujen avulla tietokantavirheilmoitus.

Servlet käyttää Etusivutekstit- ja Html-luokkien palveluja ja servletin ainoa oma metodi normaali service-metodi. Lisäksi käytetään Javan valmiita servlet- ja SQL-luokkia, jotka ilmoitettu käännösyksikön alussa.

- Etusivutekstit.java

Kirjastoluokka (liite B), jota Etusivu-luokka käyttää apuna generoidessaan etusivua. Kaikki etusivun elementit on kirjoitettu tässä luokassa html-muodossa.

Metodit:

- public static String aluste()

Metodi palauttaa sivun html-määrittelyn.

- `public static String otsikko()`  
Metodi palauttaa sivun otsikkokuvan.
- `public static String esittely()`  
Metodi palauttaa lyhyen esittelyn nuottiarkiston tarkoituksesta uusille käyttäjille.
- `public static String rekEsittely()`  
Metodi palauttaa lyhyen esittelyn rekisteröityneille käyttäjille.
- `public static String ohjeet()`  
Metodi palauttaa ohjeet aineistohakuun.
- `public static String login()`  
Metodi palauttaa sisäänkirjoittautumislomakkeen.
- `public static String hakulomake(String[] aihepiirit)`  
Metodi palauttaa nuottijulkaisujen hakulomakkeen. Parametrina annetaan merkkijonotaulukko, joka sisältää aihepiirit, jotka lisätään lomakkeeseen.
- `public static String rekLinkit()`  
Metodi palauttaa rekisteröityneille käyttäjille tarkoitetut linkit.

Jotkin edellisten metodien palauttamista html-koodeista on tallennettu luokkavakioihin.

Luokka käyttää Html-luokan palveluja.

- Login.java

Servletti (liite B) tarkistaa, onko käyttäjä rekisteröitynyt eli onko hänen sisäänkirjoittautumislomakkeessa oleva käyttäjätunnus ja salasana oikein sekä siirtää käyttäjän etusivulle.

Servletti voidaan käynnistää vain etusivulta.

parametrit:

- `ktunnus`  
Käyttäjän sisäänkirjoittautumislomakkeessa antama käyttäjätunnus.

- salasana Käyttäjän sisäänkirjoittautumislomakkeessa antama salasana.

Aluksi tarkistetaan, löytyykö käyttäjätunnus tietokannasta. Sen perusteella verrataan käyttäjän antamaa salasanaa ja tietokannassa olevaa. Jos ne täsmäävät, asetetaan käyttäjälle eväste (Cookie), jonka avulla hänet voidaan jatkossa tunnistaa rekisteröityneeksi ja joka sisältää käyttäjän käyttäjätunnuksen. Tämän jälkeen siirrytään takaisin etusivulle. Jos ilmenee ongelmia, annetaan Virheilmoitusluokan avulla oikeat virheilmoitukset.

- Logout.java

Servletti suorittaa uloskirjoittautumisen, lopetuksen.

Ei parametreja.

Servletti voidaan käynnistää vain etusivulta.

Ensin etsitään oikea eväste käyttäjän selaimesta ja tämän jälkeen kirjoitetaan samanniminen eväste, joka asetetaan häviämään heti.

### 2.1.2 Haku

Haku-servletti (liite B) suorittaa nuottijulkaisun haun etusivulla annettujen hakuehtojen perusteella ja muodostaa hakutulossivun käyttäen luokan Hakutulostekstit palveluja.

Hakutulossivulla on taulukko, jossa ovat kaikkien löytyneiden teosten perustiedot (säveltäjä, teoksen nimi, sävellysvuosi, koko ja URL-osoite) taulukossa aakkosjärjestyksessä säveltäjän nimen suhteen. Taulukon URL-osoitteet ovat linkkejä ja niistä päästään suoraan niiden osoittamille sivuille. Teosten nimet ovat sivun sisäisiä linkkejä, jotka tuovat esiin teoksen lisätiedot. Kaikista löytyneistä teoksista on lisätiedot taulukon alla taulukon mukaisessa järjestyksessä. Lisätieto-osiot muodostuvat otsikosta, joka on säveltäjän nimi ja teoksen nimi, lisätiedoista, joka on normaalia tekstiä sekä lisäyspäivästä.

Sivulta on linkit rekisteröityneiden käyttäjien toimintoihin (etusivu, nuotinlisäys, nuotinpoisto, henkilötietojen muuttaminen, lopetus), mikäli käyttäjä on rekisteröitynyt. Muuten sivulta on linkki vain etusivulle. Seuraavassa kuvataan tarkemmin toteutukseen käytetyt luokat.

- Haku.java Servletti suorittaa nuottijulkaisuhaun annettujen parametrien mukaan ja esittää tulokset.

Parameterit, jotka annetaan etusivun hakulomakkeella:

– vhakusana

Tieto siitä, onko hakusanavaihtoehto valittu. Tosi, jos arvo ei ole null.

– hakusana

Annettu hakusana. Jos tyhjä, vaihtoehtoa ei huomioida.

– vteosnimi

Tieto siitä, onko teosnimivaihtoehto valittu. Tosi, jos arvo ei ole null.

– teosnimi

Annettu teosnimihakuehto. Jos tyhjä vaihtoehtoa ei huomoida.

– saveltaja

Tieto siitä, onko säveltäjä valittu hakuehdoksi. Tosi, jos arvo ei ole null.

– etunimi

Annettu säveltäjän etunimi. Jos tyhjä, ei huomioida haussa.

– sukunimi

Annettu säveltäjän sukunimi. Jos tyhjä, ei huomioida haussa.

– vvuosi

Tieto siitä, onko sävellysvuosi valittu hakuehdoksi. Tosi, jos arvo ei ole null.

– svuosi

Annettu sävellysvuosi. Jos tyhjä, ei huomioida haussa.

- svuositarckuus

Valittu vuositarckuus, siis haetaanko annettuna vuonna sävellettyjä, vuoden jälkeen sävellettyjä vai ennen vuotta sävellettyjä teoksia. Arvot: jälkeen,tasan,ennen.

- vuutuudet

Tieto siitä, onko lisäyspäivä valittu hakuehdoksi.Tosi, jos arvo ei ole null.

- uutuudet

Valittu enimmäisaika lisäyspäivälle. Arvot: 1 kk, 2 kk,...,12kk

Haku-servletti muodostaa SQL-kyselyn annetuista parametreista ja suorittaa kyselyn. Jos kysely ei onnistu tai tietokantayhteys on poikki, antaa virheilmoituksen. Lopuksi generoi hakutulossivun Hakutulostekstit-luokan palvelujen avulla.

Luokka käyttää apuna kaikkia Hakutulosteksti-luokan metodeita ja joitakin Html-luokan palveluja. Lisäksi käytetään Javan valmiita servlet- ja SQL-luokkia, jotka ilmoitettu käännoyksikon alussa. Luokan ainoa metodi on normaali service-metodi.

- Hakutulostekstit.java

Kirjastoluokka (liite B), joka sisältää hakutulossivun luomisessa käytettävät html-tekstit. Kaikki luokan tekstit on kirjoitettu html-muodossa. Tätä luokkaa käyttää ainoastaan Haku-servletti.

Luokan metodit:

- public static String alkutekstit() Palauttaa hakutulossivun alkutekstit: dokumentin html-määrittelyn ja sivun otsikon.

Ei parametreja.

- public static String taulukko(boolean alku)

Palauttaa html-tilukkoalusteen, jos parametri alku on tosi, muuten html-tilukon lopetuksen.

- `public static String sarakenimet()`

Palauttaa hakutulossivun tulostaulukon sarakkeiden nimet html-muodossa.

- `public static String tulosrivi(String sukunimi, String etunimi, String teos, String svuosi, String koko, String url, int hakunro)`

Palauttaa html-muodossa yhden taulukon rivin annetuista parametreista. Suku- ja etunimi ovat säveltäjän nimi, teos on löydetyn teoksen nimi, svuosi on sävellysvuosi, koko on teoksen koko sivuina, url on teoksen URL- osoite ja hakunro on tieto siitä, monentenako rivi on taulukossa (käytetään sivun sisäisessä linkissä lisätietoihin).

Palautteessa teoksen nimeen on lisätty sivun sisäinen linkki parameterina annettuun hakunumeroon (hakunro) ja URL-osoitteesta on tehty html-hyperlinkki.

- `public static String lisätiedot(String otsikko, String tieto, String pvm, int nro)`

Palauttaa teoksen lisätieto-osion. Otsikko on nimetty annetulla numerolla, jotta siihen voidaan tehdä sivun sisäinen linkki. Otsikko tulee sellaisenaan osion otsikoksi ja pvm tulee sellaisenaan teoksen lisäyspäivämääräksi.

- `public static String linkit(boolean rekisteroitynyt)`

Palauttaa hakutulossivun linkit. Jos parametri rekisteroitynyt on tosi, palauttaa linkit kaikkiin rekisteröityneiden käyttäjien toimintoihin (etusivu, nuottilisäys ja -poisto, henkilötietojen muuttaminen ja lopetus) muuten palauttaa linkin ainoastaan etusivulle.

Luokka käyttää Html-luokan palveluja. Ainoa luokkaa käyttävä servletti on Haku.

### 2.1.3 Nuotin lisäys arkistoon

Nuottilomake-servletti generoi uuden nuottilisäyslomakkeen käyttäen apuna Nuottilomaketekstit-luokan palveluja. Lomakkeen käsittelee Nuottilisäys- servletti, joka

myös antaa palautteen onnistumisesta.

Nuotinisäyslomakkeessa (liite 4) on kentät teoksen nimelle, säveltäjän nimille, aihepiireille, uuden aihepiirin lisäykselle, teoksen koolle, sävellysvuodelle ja URL-osoitteelle. Näistä tiedoista on pakko antaa ainakin teoksen nimi ja URL-osoite.

Nuotinisäyslomakkeesta on linkit kaikkiin rekisteröityneiden käyttäjien toimintoihin.

Seuraavana on esitelty tarkemmin toteutuksessa käytetyt luokat.

- Nuottilomake.java

Servletti (liite B) generoi nuotinisäyslomakkeen käyttäen apuna Nuottilomaketekstit-luokan palveluja.

Ei parametreja.

Ensin tarkistetaan, onko käyttäjä rekisteröitynyt, sitten haetaan tietokannasta sen hetkiset tietokannassa olevat aihepiirit ja niiden perusteella tuotetaan nuotinisäyssivu.

Luokka käyttää Nuottilomaketekstit- Html- ja valmiita Javan käännösyksikön alussa ilmoittamia luokkia ja ainoa luokan metodi on normaali service-metodi.

- Nuottilomaketekstit.java

Kirjastoluokka (liite B) nuotinisäyssivun generoimiseen. Kaikki luokan tekstit ovat html-muodossa.

Metodit:

- public static String aluste()

Palauttaa nuotinisäyssivun html-alusteen sekä otsikon ja ohjeet.

Ei parametreja.

- public static String lomake(String[] aihepiirit)

Palauttaa nuotinisäyslomakkeen annetuilla aihepiireillä. Aihepiirit tulevat lomakkeeseen monivalittavaksi listaksi.

- `public static String linkit()`  
Palauttaa linkit kaikkiin rekisteröityneiden käyttäjien toimintoihin.
- `public static String lisattyKantaan()`  
Palauttaa täydellisen html-sivun, jossa ilmoitetaan, että nuottijulkaisu on onnistuneesti lisätty nuottiarkistoon.
- `private static String lomakeA()`  
Luokan sisäinen metodi joka palauttaa nuotinisäyslomakkeesta ensimmäiset aina samat osat, siis osat ennen muuttuvaa aihepiiri-osaa.
- `private static String lomakeB()` Luokan sisäinen metodi joka palauttaa nuotinisäyslomakkeesta jälkimmäiset aina samat osat, siis osat muuttuvan aihepiiri-osan jälkeen.

Luokka käyttää Html-luokan palveluja. Luokka käyttävät sekä Nuottilomake-että Nuotinisays-servletti.

- Nuotinisays.java

Servletti (liite B) nuotinisäyslomakkeen käsittelemiseksi.

Parametrit, jotka on annettu nuotinisäyslomakkeessa (määritelty luokassa Nuottilomaketekstit.java):

- nimi  
Teoksen nimi.
- etunimi  
Säveltäjän etunimi.
- sukunimi  
Säveltäjän sukunimi.
- sävellysvuosi  
Teoksen sävellysvuosi.



- aihepiiri  
Uusi lisätty aihepiiri.
- koko  
Teoksen koko sivuina.
- url  
Teoksen URL-osoite.
- lisätiedot  
Teoksen lisätiedot.
- vaihepiiri  
Teokseen liittyvät aihepiirit. Muoto poikkeuksellisesti String- taulukko.

Servletti tarkistaa, että teoksen nimi ja URL-osoite on annettu. Sitten lisätään teoksen säveltäjä tietokantaan, jos se ei vielä ole siellä. Sitten lisätään teos, jos ei vielä täsmälleen samaa ole jo lisätty. Lopuksi asetetaan aihepiirit kohdalleen tietokantaan ja tarvittaessa lisätään uusi annettu aihepiiri. Jos ongelmia ilmenee, annetaan virheilmoitukset, muuten annetaan Nuottilomaketekstit-luokan avulla tiedotus nuotin lisäyksen onnistumisesta.

Servletti voidaan käynnistää vain nuottilisäyssivulta.

#### 2.1.4 Nuotin poisto

Nuotin poistosivu (liite 5) generoituu eri tavoilla rekisteröityneille käyttäjille ja arkiston hoitajalle. Rekisteröityneiden käyttäjien nuotinpoistolomakkeessa ei näy nuotien lisääjän käyttäjätunnusta ja jokaiselle käyttäjälle näkyvät siis vain itse lisäämät nuottijulkaisut. Nuottijulkaisun tietoina on säveltäjä, teoksen nimi, URL- osoite ja lisäyspäivä. Jokaisen nuottijulkaisurivin edessä on valintalaatikko, josta nuottijulkaisu voidaan valita poistettavaksi.

Arkiston hoitajalle sivu generoituu siten (liite D), että siinä näkyvät kaikkien käyttäjien lisäämät nuottijulkaisut. Lisäksi nuottijulkaisurivillä on myös lisääjän käyttäjätunnus.

Se toimii sivun sisäisenä linkkinä käyttäjän tietoihin, jotka ovat omana taulukkonaan sivun lopussa.

Sivulta on linkit kaikkiin rekisteröityneiden käyttäjien toimintoihin.

Nuotinpoistolomakkeen generoi Nuotinpoistolomake-servletti käyttäen apunaan Nuotinpoistotekstit-luokkaa. Lomakkeen käsittelee Nuotinpoisto-servletti. Näistä seuraavassa tarkemmat esittelyt.

- Nuotinpoistolomake.java

Servletti (liite B) generoi nuotinpoistolomakkeen, josta voidaan valita poistettavat nuottijulkaisut.

Ei parametreja.

Servletti hakee ensin tiedon tietokannasta, onko käyttäjä arkiston hoitaja. Sen jälkeen haetaan tietokannasta tiedot joko käyttäjän evästeestä saatavalla käyttäjätunnuksella lisätyistä teoksista (rekisteröitynyt käyttäjä) tai kaikkien nuottiarkiston teosten tiedot (arkiston hoitaja). Jos käyttäjä on arkiston hoitaja lisätään teosten tietoihin lisääjä ja tehdään teostietotaulukon perään taulukko kaikista arkiston käyttäjistä ja heidän tiedoistaan.

Servletti käyttää apuna luokan Nuotinpoistotekstit palveluja ja käyttää sekä luokkaa Html että javan valmiita käännösyksikössä ilmoitettuja servlet- ja SQL-luokkia.

- Nuotinpoistotekstit.java

Kirjastoluokka (liite B) nuotinpoistosivun generoimiseen tarvittaville html-teksteille. Luokka tarjoaa valmiita html-koodilla tehtyjä sivunosia, joista Nuotinpoistolomake-servletti voi koota oikeanlaisen nuotinpoistolomakkeen.

Metodit:

- public static String aluste()

Palauttaa nuotinpoistosivun html-määrittelyn ja sivun otsikon.

– `public static String napit()`

Palauttaa nuotinpoistolomakkeen tyhjennys ja lähetysnapit.

– `public static String lomake()`

Palauttaa nuotinpoistolomakkeen alusteen.

– `public static String lomakeotsikko(boolean ktunnuksella)`

Palauttaa nuotinpoistolomakkeen taulukon otsikot. Jos parametri ktunnuksella on tosi, lisätään viimeiseksi sarakkeeksi käyttäjätunnussarake.

– `public static String teosrivi(String sukunimi, String etunimi, String teosnimi, String lisaysaika, String lisaaja, int tunniste, String url)`

Palauttaa yhden rivin nuotinpoistolomakkeeseen. Parametrit sukunimi ja etunimi ovat säveltäjän nimi, teosnimi teoksen nimi, lisaysaika teoksen lisäysaika arkistoon, tunniste on tietokannan antama teoksen tunnistenumero, url teoksen URL-osoite ja lisaaja lisääjän käyttäjätunnus. Jos lisääjää ei ole, ei lisätä koko saraketta. Jos lisääjä on, siihen lisätään sivun sisäinen linkki samaan käyttäjätunnukseen.

– `public static String kayttajatietotaulukko()`

Palauttaa käyttäjätietotaulukon alusteen ja taulukko-otsikot.

– `public static String kayttajatietorivi(String ktunnus, String sukunimi, String etunimi, String sposti)`

Palauttaa yhden rivin käyttäjätietotaulukkoon annetuilla parametreilla. Ktunnus on käyttäjän käyttäjätunnus, suku- ja etunimi käyttäjän nimi ja sposti käyttäjän sähköpostiosoite. Käyttäjätunnus nimetään käyttäjätunnukseksi, jotta siihen voidaan tehdä sivun sisäinen linkki.

– `public static String lomakelopuke()`

Palauttaa lomakkeen lopetukseen tarvittavat html-tagit.

– `public static String linkit()`

Palauttaa linkit kaikkiin rekisteröityneiden käyttäjien toimintoihin.

Luokka käyttää Html-luokan palveluja. Luokkaa käyttää vain Nuotinpoistolomake-servletti.

- Nuotinpoisto.java

Servletti (liite B) nuottijulkaisun poistamiseen nuottiarkistosta.

Parametrit, jotka saatu nuotinpoistolomakkeesta: parametrien niminä kaikkien poistettavien nuottijulkaisujen tunnistenumerot, siis parametrien nimet ja määrä vaihtelevat riippuen käyttäjän valitsemista nuottijulkaisuista.

Servletti muodostaa tietokannan päivityslauseen annettujen parametrien perusteella ja suorittaa kyselyn. Tämän jälkeen servletti ohjaa käyttäjän uudestaan Nuotinpoistolomake-servletin generoimalle nuotinpoistosivulle.

Servletti käyttää käännoyksikössä ilmoitettuja javan valmiita servlet- ja SQL-luokkia.

### **2.1.5 Rekisteröityminen ja henkilötietojen muuttaminen**

Rekisteröitymissivun generoi Rekisterointilomake-servletti käyttäen hyväksi Rekisterointitekstit-luokan palveluja. Lomakkeen käsittelee Rekisterointi-servletti.

Sivu generoituu eri tavoin rekisteröidyille ja rekisteröitymättömille käyttäjille. Rekisteröitymättömille käyttäjille generoidaan rekisteröitymislomake (liite 7), jossa on lyhyet ohjeet ja tyhjä lomake henkilötietoja varten. Rekisteröityneille käyttäjille haetaan tietokannasta sen hetkiset käyttäjätiedot lomakkeeseen (liite 8). Tietenkin sivun otsikko on tällöin Henkilötietojen muuttaminen ja ohjeet koskevat henkilötietojen muuttamista. Muutoslomakkeessa ei voi muuttaa käyttäjätunnusta.

Rekisteröitymissivulta on linkki ainoastaan etusivulle mutta henkilötietojen muutosivulta on linkit kaikkiin rekisteröityneiden käyttäjien toimintoihin.

Sivujen toteutuksessa käytettävistä luokista on annettu seuraavassa tarkempi selonteke.

- Rekisterointilomake.java Servletti (liite B) rekisteröinti- tai henkilötietolomakkeen generoimiseksi.

Ei parametreja.

Ensin tarkistetaan, onko käyttäjällä tunniste-evästettä eli onko käyttäjä rekisteröitynyt ja sisäänkirjoittautunut. Jos ei, generoidaan rekisteröitymissivu, jossa on tyhjä lomake henkilötietoja varten ja ohjeet sen täyttämiseksi. Jos käyttäjällä on tunniste-eväste, katsotaan sen sisältämä käyttäjätunnus ja sen perusteella haetaan henkilön käyttäjätiedot tietokannasta. Generoidaan saaduilla tiedoilla esitetyt rekisteröitymislomake, jossa käyttäjätunnusta ei voi vaihtaa.

Luokka käyttää Rekisterointitekstit-luokan palveluja sekä valmiita käännoyksikössä ilmoitettuja java servlet ja SQL luokkia.

- Rekisterointitekstit.java

Kirjastoluokka (liite B) rekisteröintisivun ja henkilötietojen muuttamissivujen generoimiseen. Luokka sisältää kaikki kyseisten sivujen tekstit html-muodossa.

Metodit:

- public static String aluste(boolean muutoslomake)

Jos parametri muutoslomake on tosi, palauttaa henkilötietojen muutoslomakesivun html-alusteen ja sivun otsikon muuten rekisteröitymissivun vastaavat.

- public static String ohjeet(boolean muutoslomake)

Jos parametri muutoslomake on tosi, palauttaa ohjeet henkilötietojen muuttamiseen, muuten ohjeet rekisteröitymiseen.

- public static String reklomake()

Palauttaa tyhjän rekisteröintilomakkeen html-koodina.

- public static String muutoslomake(String eNimi, String sNimi, String email, String kTunnus, String sSana)

Palauttaa henkilötietojen muutoslomakkeen valmiiksi esitäytettynä annetuilla parametreilla, html-koodina.

– `public static String linkit(boolean muutoslomake)`

Jos parametri `muutoslomake` on tosi, palauttaa linkit kaikkiin rekisteröityneiden käyttäjien toimintoihin, muuten linkin etusivulle.

Luokka käyttää `Html`-luokan palveluja.

- `Rekisterointi.java`

Servletti (liite B) rekisteröinti- ja henkilötietojen muutoslomakkeiden käsittelemiseen.

Parametrit, jotka saatu rekisteröinti- tai henkilötietojen muutoslomakkeesta:

– `etunimi`

Käyttäjän etunimi.

– `sukunimi`

Käyttäjän sukunimi.

– `sposti`

Käyttäjän sähköpostiosoite.

– `ktunnus`

Käyttäjän käyttäjätunnus.

– `salasana1`

Käyttäjän antama salasana.

– `salasana2`

Käyttäjän antama salasanan varmistus.

– `poisto`

Tämä parametri saadaan vain henkilötietojen muutoslomakkeesta. Parametri ilmoittaa, onko kysymyksessä henkilötietojen poisto. Jos parametri

ei ole null poistetaan kyseisen käyttäjän kaikki tiedot ja lisäämät teokset tietokannasta.

Ensin servletti tarkistaa evästeestä, onko käyttäjä rekisteröitynyt. Jos ei ole, suoritetaan annettujen tietojen lisäys tietokantaan ja siirretään käyttäjä etusivulle. Jos käyttäjä on rekisteröitynyt, suoritetaan tietokannan päivitys annetuilla tiedoilla ja siirrytään etusivulle. Molemmissa tapauksissa ensin tarkastetaan, että sukunimi ja käyttäjätunnus on annettu ja että salasana1 ja salasana2 ovat täsmälleen samat. Tarvittaessa annetaan virheilmoitukset virheelisistä tai puuttuvista arvoista tai tietokantayhteyden katkeamisesta.

Luokka käyttää Rekisterointitekstit-, Html-, Virheilmoitukset- ja valmiiden käännösyksikössä ilmoitettujen java servlet- ja SQL-luokkien palveluita.

### 2.1.6 Apuluokat

Apuluokat ovat sekava kokoelma eri luokkien käyttämiä apuluokkia. Tietokantaluokassa määritellään käytettävä tietokanta ja Virheilmoitukset ovat erilaisten virheilmoitusten generoimista varten. Html-luokasta saadaan html-koodia. Seuraavassa luokista tarkemmat esittelyt.

- Tietokanta.java

Kirjastoluokka (liite B) tietokanta-asetuksien tallentamiseen sekä tietokantayhteyden avaamiseen ja sulkemiseen. Tämän luokan luokkavakioita muuttamalla voidaan vaihtaa käytettävää tietokantaa.

Metodit:

- public static String tkAjuri()  
Palauttaa käytettävän tietokantaajurin.
- public static String tkServeri()  
Palauttaa käytettävän tietokantaserverin.

- `public static String tkKayttaja()`  
Palauttaa käytettävän tietokannan käyttäjätunnuksen.
- `public static String tkSalasana()`  
Palauttaa käytettävän tietokannan salasanan.
- `public static Connection luoYhteys()`  
Palauttaa yhteys-tyyppisen olion, jonka avulla voidaan käyttää tietokantaa, eli saadaan siihen yhteys.
- `public static void suljeYhteys(Connection yhteys)`  
Sulkee parametrina annetun yhteyden tietokantaan.

- Virheilmoitukset.java

Kirjastoluokka (liite B) virheilmoitusten generoimiseen. Luokan metodit palauttavat valmiin html-sivun, joka sisältää tietyn virheilmoituksen.

- `public static String tkYhteysvirhe()`  
Palauttaa virheilmoituksen siitä, että tietokantaan ei saada yhteyttä.
- `public static String SQLVirhe(String virhe)`  
Palauttaa virheilmoituksen ohjelmassa tapahtuneesta yllättävästä virheestä. Tässä virheilmoituksessa on mukana parametrina annettu tietokannan antama virheilmoitus sekä luokassa määritelty ohjelman toimittajan osoitetiedot, joihin pitäisi ottaa yhteyttä.
- `public static String eiNumero()`  
Palauttaa virheilmoituksen virheellisestä arvosta.
- `public static String salasanavirhe()`  
Ilmoittaa, että salasanan varmistus on ollut väärä.
- `public static String ktunnus()`  
Palauttaa virheilmoituksen siitä, että käyttäjätunnusta ei ole annettu.
- `public static String ktunnusLaiton()`



Palauttaa virheilmoituksen siitä, että annettu käyttäjätunnus on sisältänyt sopimattomia merkkejä, esim. hipsuja.

– `public static String sukunimi()`

Palauttaa virheilmoituksen siitä, että sukunimi puuttuu lomakkeesta.

– `public static String eiPipareita()`

Palauttaa virheilmoituksen siitä, että käyttäjän selain ei tue evästeiden käyttöä.

– `public static String login()`

Palauttaa virheilmoituksen siitä, että sisäänkirjoittautuminen ei ole onnistunut virheellisen käyttäjätunnuksen tai salasanan vuoksi.

– `public static String eiNuottitietoja()`

Palauttaa virheilmoituksen siitä, että nuotinisäyslomakkeesta puuttuu vaadittavia tietoja.

– `public static String kielletytMerkit()`

Palauttaa virheilmoituksen siitä, että annettu syöte sisältää sopimattomia merkkejä, esim. hipsuja.

– `private static String virheilmoitus(String virheotsikko, String virheselitys, String SQLvirhe)`

Luokan sisäinen metodi, joka generoi varsinaisen virheilmoitussivun annetuilla parametreilla.

Luokka käyttää Html-luokan palveluja.

- `Html.java`

Kirjastoluokka (liite B) html-koodin kirjoittamisen helpottamiseksi.

Metodit:

– `public static String aluste(String sivun_otsikko)`

Palauttaa html-dokumentin määrittelyn annetulla sivuotsikolla.

- `public static String aluste(String sivun_otsikko, Stringtausta)`  
Palauttaa html-dokumentin määrittelyn annetulla sivuotsikolla sekä annetulla taustakuvalla.
- `public static String lopetus()`  
Palauttaa html-dokumentin lopetuksen lopputagit.
- `public static String otsikkokuva(String teksti, String kuva)`  
Palauttaa html-koodin kuvan liittämiseksi dokumenttiin. Vaihtoehtotekstiksi tulee annettu teksti.
- `public static String otsikko(String otsikko, int koko)`  
Palauttaa html-otsikon annetun kokoisena. Koko välillä 1-9.
- `public static String linkki(String otsikko, String osoite)`  
Palauttaa html-linkin annettuun osoitteeseen ja nimeää linkin annetulla otsikolla.
- `public static String email(String otsikko, String osoite)`  
Palauttaa html-linkin annettuun sähköpostiosoitteeseen ja nimeää linkin annetulla otsikolla.
- `public static String nimea(String nimettävä, String nimi)`  
Nimeää nimettävän annetulla nimellä, jotta nimettävään voidaan tehdä sivun sisäinen linkki.
- `public static String taulukkoaluste(int reunuskoko, String tyyli)`  
Palauttaa taulukkoalusteen annetulla reunuskoolla ja annetulla tyyllillä. Tyyli annetaan kokonaisena tyylimäärittely (style) lauseella.
- `public static String taulukkolopetus()`  
Palauttaa taulukon lopputagin.
- `public static String taulukkorivi(boolean aloitus)`  
Palauttaa taulukkorivin alkutagin jos parametri aloitus on tosi, muuten lopputagin.

- `public static String taulukkoalkio(String alkio)`  
Palauttaa taulukkoalkiona annetun merkkijonon.
- `public static String kappale(String kappale)`  
Palauttaa html-kappaleena annetun merkkijonon.

## 3 Käyttöohje

Tämä osa antaa tarkat ohjeet siitä, miten ohjelma asennetaan, käynnistetään ja miten ohjelmaa käytetään. Tosin ohjelma on pyritty tekemään mahdollisimman arvatavaksi, jotta varsinaisia käyttöohjeita ei tarvittaisi. Lisäksi ohjelman sivuilta löytyy lyhyet käyttöohjeet eri toimintojen käyttämiseksi.

### 3.1 Ohjelman asennus

Koska ohjelma on tietokantasovellus, on ensin vietävä ohjelman käyttämät taulut tietokantaan. Sinulla on oltava pääsy johonkin tietokantaan ja jonkinlainen tietokantaliittymä ajaa suunnitteludokumentin liitteenä olevat `create table` sekä tämän dokumentin liitteenä olevat `create sequence`-lauseet (liite C) tietokantaan.

Esimerkiksi tämä ohjelma on alunperin asennettu Helsingin yliopiston atk-keskuksen Oracle-tietokantaan. Tällöin asentaminen tapahtui siirtämällä ajettavat tiedostot palvelimelle ja käyttämällä tietokantaliittymää `sqlplus`. Siis kokonaisuudessaan komento jolla taulut ajettiin tietokantaan oli `sqlplus / @ctable.sql`. Tämän lisäksi oli ajettava tiedosto `sequence.sql` tietokantaan. Tämä tapahtui niin ikään komennolla `sqlplus / @sequence.sql`.

Tämän lisäksi kannattaa ajaa arkistonhoitajan käyttäjätiedot tietokantaan. Ohjelman mukana toimitetaan tiedosto `hoitaja.sql` (liite C). Se sisältää arkiston hoitajan luomiseksi tarvittavat SQL-komennot. Tiedostoa kannattaa muuttaa siten, että muuttaa

salasanaksi haluamansa sanan. Myös muita tietoja voi muuttaa. Tämän jälkeen voi tiedoston ajaa tietokantaan kuten edellä on tehty muillekin tiedostoille.

Ohjelman suoritukseen tarvittavat luokat voidaan sijoittaa kaikki samaan hakemistoon. Suositeltavaa kuitenkin on, että servletit sijoitettaisiin omaan hakemistonsa ja apuluokat muualle. Tämä kuitenkin edellyttää hakemistojen lisäämistä Java-virtuaalikoneen ympäristömuuttujaan. Käytännössä on siis helpompaa sijoittaa kaikki tiedostot siihen hakemistoon, jossa palvelimen omistaja on ilmoittanut servletejä säilytettävän.

Alkuperäisessä toteutuksessa tiedostot ovat palvelimen db.cs.helsinki.fi hakemistossa `/home/asyreeni/jserv/servlets`

Ohjelman tarvitsemat kuvat on sijoitettava johonkin muuhun hakemistoon, jossa ne ovat näkyvillä netissä. Jokaisen ohjelman tekstiluokan luokkamuuttujaa tausta on muutettava siten, että sen arvoksi annetaan täydellinen hakemistopolku kuvat sisältävään hakemistoon. Muista antaa hakemistoille oikeudet. Lisäksi on Etusivutekstit-luokan luokkamuuttujaan otsikkokuva vaihdettava tuo hakemistopolku.

Alunperin kuvat on sijoitettu hakemistoon `http://db.cs.helsinki.fi/u/asyreeni`.

Ennen kooditiedostojen kääntämistä on Tietokanta-luokassa vaihdettava käytettävä tietokantaserveri, tietokanta-ajuri ja käyttäjän nimi sekä salasana käytössä oleviin arvoihin.

Alunperin tietokantana on käytetty Helsingin yliopiston tietokantaa, siis tietokantaserveri vakio on ollut: `jdbc:oracle:thin:@kontti.helsinki.fi:1522:tktb` Tietokanta-ajurin on oltava tietokantaan sopiva java-ajuri. Alunperin on käytetty Oraclen JDBC-ajuria, siis luokka vakio on ollut: `oracle.jdbc.driver.OracleDriver`.

Sen palvelimen, jossa java-servletejä, siis tätä ohjelmaa käytetään, asetuksiin on mainittava käytettävä tietokanta-ajuri. Esimerkiksi tämä ohjelma asennettiin db.cs.-helsinki.fi -palvelimelle ja tuolla palvelimella tietokanta-ajuri määriteltiin tiedostossa `environment`.

Näiden toimenpiteiden jälkeen kooditiedostot voidaan kääntää class- tiedostoiksi, jos java-kääntäjä löytää tarvittavat java-servlet luokat.

Tämän jälkeen java-virtuaalikone on käynnistettävä. Alkuperäisessä toteutuspalvelimessa oli laadittu valmiit komentosarjat koneen käynnistämiksi. Siksi onkin palvelimen ylläpitäjältä pyydetty ohjeet siitä, miten kyseisessä koneessa servletit käännetään ja virtuaalikone käynnistetään.

Näiden toimenpiteiden jälkeen Nuottiarkiston pitäisi olla käytettävissä www-pohjaisella selaimella.

Koska kaikki asennusohjeet perustuvat käytössä oleviin palvelimiin ja tietokantaan on asentamiseen liittyvät yksityiskohdat selvitettävä kussakin tapauksessa erikseen.

## **3.2 Ohjelman käynnistys ja käyttö**

Kun sovellus on oikein asennettu Nuottiarkisto käynnistyy asettamalla web- selaimessa osoitteeksi se hakemistopolku josta asennuspalvelimen ylläpitäjä on ilmoittanut servlet-sovellusten käynnistyvän ja tiedostonimeksi Etusivu.

Esimerkiksi tämä ohjelma alkuperäisessä toteutuksessa käynnistyy osoitteella <http://db.cs.helsinki.fi/s/asyreeni/Etusivu>.

Seuraavassa on annettu käyttöohjeet kaikkien nuottiarkiston toimintojen käyttämiseen.

### **3.2.1 Nuottijulkaisujen haku arkistosta**

Etusivulla (liite 2) on lomake, jonka avulla voi suorittaa hakuja nuottiarkistoon. Lomakkeeseen annetaan ne tiedot, joiden perusteella haetaan. Jokainen hakuvaihtoehto, joka halutaan hakuperusteeksi, on merkittävä lomakkeen vasemmassa reunassa olevalla rastilla. Useita hakuvaihtoehtoja voidaan merkitä kerralla hakuvaihtoehdoiksi, jolloin haettavien nuottijulkaisujen on täytettävä kaikki merkityt hakukriteerit.

Mikäli hakuperuste on merkitty, mutta sille ei ole annettu mitään arvoja, hakuperustetta ei huomioida haussa.

Täytettyäsi hakulomakkeen suorita haku painamalla lomakkeen alhaalla olevaa ”Lähetä-nappia. Voit myös tyhjätä lomakkeen ”Tyhjennä-napista.

Seuraavassa on esitelty kaikki hakuperusteet ja niiden käyttö.

## 1. hakusana

Jos merkitset tämän hakuperusteen ja annat jonkin sanan, nuottiarkistosta haetaan kaikki ne nuottijulkaisut, joiden säveltäjän nimessä, teoksen nimessä, URL-osoitteessa, aihepiireissä tai lisätiedoissa esiintyy antamasi sana.

Antamasi sanan kirjainkoolla ei ole merkitystä ja hakusanan ei tarvitse olla kokonainen sana ja hakusana voi sijaita nuottijulkaisun tiedoissa myös sanan keskellä. Niinpä esimerkiksi antamalla hakusanaksi kirjain a, saadaan hakutulokseksi kaikki ne teokset, joiden edellä luetelluissa tiedoissa a-kirjain esiintyy. Hakutuloksia voi siis rajoittaa antamalla pitempiä hakusanoja.

Hakusana ei saa sisältää hipsuja. Hipsujen antaminen hakusanassa aiheuttaa virheilmoituksen.

Esimerkki:

Annat antaa hakusanaksi 'piano' ja saat hakutulokseksi kaikki teokset, joiden tiedoissa sana piano esiintyy.

## 2. teoksen nimi

Jos valitset tämän hakuperusteen, saat hakutulokseksi kaikki ne nuottijulkaisut, joiden nimi alkaa antamallasi hakusanalla. Kirjainkoolla ei ole merkitystä ja antamalla pitempiä hakusanoja rajoitat löydettyjen nuottijulkaisujen määrää.

Teoksen nimi ei saa sisältää hipsuja. Hipsujen antaminen aiheuttaa virheilmoituksen.

Esimerkki:

Annat hakusanaksi konsertto ja saat hakutulokseksi kaikki konsertto-alkuiset nuottijulkaisut, esimerkiksi ”Konsertto pianolle” ja ”Konsertto sellolle”.

### 3. säveltäjä

Valitsemalla säveltäjän hakuperusteeksi saat tulokseksi ne nuottijulkaisut, joiden säveltäjän nimet alkavat antamillasi sanoilla.

Ylemmään kenttään annat säveltäjän etunimen (tai sen ensimmäiset kirjaimet) ja alempaan säveltäjän sukunimen (tai sen ensimmäiset kirjaimet). Huomioi, että jotkin selaimet eivät näytä etu- ja sukunimen syöttökenttien otsikoita, mutta järjestys on sama.

Voit antaa joko sekä etu- että sukunimen tai vain jomman kumman hakuperusteeksi jättämällä toisen kentän tyhjäksi. Antamasi sanojen kirjainkoolla ei ole merkitystä haun tulosten kannalta.

Säveltäjän nimet eivät saa sisältää hipsuja. Hipsujen antaminen aiheuttaa virheilmoituksen.

Esimerkki:

Valitset hakuehdoksi säveltäjän, annat säveltäjän sukunimeksi kirjaimen a, jolloin saat hakutuloksen kaikista nuottijulkaisuista, joiden säveltäjän sukunimen ensimmäinen kirjain on a.

### 4. sävellysvuosi

Antamalla hakuehdoksi sävellysvuoden voit määritellä vuoden, jolloin haettavat teokset on sävelletty. Anna vuosi kokonaisena (esim. 1980) ja ole huolellinen, että annat vain numeroita.

Voit myös valita alemmasta valikosta, haetaanko antamanasi vuonna sävelletyt (vaihtoehto ”tasan”), antamasi vuoden jälkeen sävelletyt (vaihtoehto ”jälkeen”) tai ennen antamaasi vuotta sävelletyt (vaihtoehto ”ennen”).

Esimerkki:

Annat sävellysvuodeksi 1980 ja valitset alapuolen listasta vaihtoehdon ”jälkeen”. Tällöin saat hakutulokseksi kaikki ne sävellykset, jotka on sävelletty vuoden 1980 jälkeen.

#### 5. aihepiiri

Voit valita listalta aihepiirin, johon haettavat nuottijulkaisut kuuluvat. Ota huomioon, että sama nuottijulkaisu voi kuulua useampaankin aihepiiriin. Muista merkitä hakuvaihtoehto rastilla valituksi.

Esimerkki:

Valitset aihepiiriin ”moderni” ja saat hakutuloksen kaikkien moderniksi luokiteltujen teosten tiedoista.

#### 6. vain uutuudet

Kun valitset tämän vaihtoehdon, voit valita listasta, kuinka monta kuukautta sitten nuottiarkistoon lisättyjä haettavat julkaisut saavat enintään olla. Vaihtoehdot ovat 1-12kk. Päivämäärät eivät välttämättä ole täysin tarkkoja.

Esimerkki:

Valitset listasta vaihtoehdon 1kk ja merkitset hakuperusteen rastilla. Tällöin saat hakutulokseksi kaikki ne julkaisut, jotka on lisätty arkistoon viimeisen kuukauden aikana.

### 3.2.2 Hakutulosten katselu

Lähetettyäsi etusivun hakulomakkeen asianmukaisesti täytettynä saat eteesi hakutulossivun (liite 9). Sivun alussa on taulukko, jossa ovat hakuperusteesi vastaavat teokset aakkosjärjestyksessä. Jokaisesta teoksesta näkyy säveltäjän nimi, teoksen nimi, sävellysvuosi, koko ja URL-osoite.

URL-osoite toimii normaalina html-linkkinä, josta pääset suoraan nuottijulkaisun luokse. Teoksen nimi puolestaan toimii linkkinä teoksen lisätietoihin, jotka ovat hakutulossivun loppuosassa. Lisätietojen otsikoina ovat säveltäjien nimet sekä kyseiset



teokset. Lisätietoihin on myös merkkitty ajankohta, jolloin teokset on lisätty nuot-tiarkistoon.

Jos olet rekisteröitynyt (kts. 3.2.3), hakutulossivulla on linkit etusivulle, nuotinpoisto-sivulle, nuotinisäyssivulle sekä lopetukseen. Muuten sivulta on linkki vain etusivulle.

### **3.2.3 Rekisteröityminen**

Etusivulla (liite 2) on sekä esittelyssä, että sisäänkirjoittautumisen yhteydessä linkki rekisteröitymissivulle. Rekisteröitymissivulla on lomake, jolla annat henkilötietosi.

Sinun on annettava ainakin sukunimesi ja käyttäjätunnuksesi. On kuitenkin erittäin suositeltavaa, että annat kaikki pyydetyt tiedot. Joissain epäilyttävissä tapauksissa saattaa arkiston hoitaja ottaa sinuun yhteyttä ennen kuin poistaa nuottijulkaisujasi arkistosta.

Salasanoja ei ole pakko antaa, mutta se on erittäin suositeltavaa. Kun annat salasa-nan, anna se molempiin salasanakenttiin. Salasanojen on oltava täsmälleen samat.

Kun olet täyttänyt lomakkeen, lähetä se ”Lähetä-painikkeella. Voit myös tyhjentää lomakkeen ”Tyhjennä-painikkeella.

Huomaa, että missään kohtaa lomaketta ei hyväksytä hipsuja.

Muista ennen rekisteröitymistä tarkistaa, että selaimesi hyväksyy evästeiden käytön. Jos et suostu siihen, että selain asentaa tilapäisen tunnistustiedoston (Cookie,eväste) koneellesi, et voi rekisteröityä.

### **3.2.4 Sisäänkirjoittautuminen**

Etusivulla (liite 2) on hakulomakkeen alla pieni lomake, jossa pyydetään käyttäjätunnus ja salasana. Anna käyttäjätunnuksesi ja salasanasi oikeisiin kenttiin ja paina ”Lähetä-nappia. Jos antamasi tiedot ovat oikein ja olet rekisteröitynyt (kts. 3.2.3) pääset re-

rekisteröityneiden käyttäjien etusivulle. Älä hämäännä, rekisteröityneiden käyttäjien etusivulla ja tavallisessa etusivulla ei ole mitään eroa, kun että rekisteröityneiden käyttäjien etusivulla esittelyosuus on hieman erilainen ja sivulla on linkit rekisteröityneiden käyttäjien toimintoihin.

Jos saat virheilmoituksen siitä, että salasanasi tai käyttäjätunnuksesi on ollut väärä, yritä uudelleen.

### **3.2.5 Henkilötietojen muuttaminen**

Kun olet rekisteröitynyt ja sisäänkirjoittautunut ja olet rekisteröityneiden käyttäjien etusivulla, voit valita linkeistä vaihtoehdon ”Henkilötietojen muuttaminen”. Tällöin saat eteesi samannäköisen lomakkeen kuin rekisteröityessäsi sait, henkilötietojen muutoslomakkeen (liite 8). Kuitenkin käyttäjätunnusta et pysty muuttamaan. Lomake on valmiiksi täytetty nykyisillä tiedoillasi ja voit muuttaa kaikkia muutettavissa olevia tietoja.

Huom! Laittamalla rastin ”Henkilötietojen poisto-vaihtoehtoon poistat tietosi ja lisäämäsi teokset arkistosta (kts. kohta 3.2.6).

Edelleenkään lomakkeesi ei saa sisältää hipsuja.

Korjaa tarpeelliset tiedot ja lähetä lomake ”Lähetä-painikkeesta. Voit myös palauttaa lähtöarvot painamalla ”Tyhjennä-painiketta. Huomaa, että lähettäessäsi on lomakkeessa oltava ainakin sukunimesi.

### **3.2.6 Henkilötietojen poistaminen nuottiarkistosta**

Henkilötietojen poistaminen tapahtuu henkilötietojen muutossivulta (liite 8, kts 3.2.5). Henkilötietojen muutossivulla on kohta ”Henkilötietojen poisto”. Kun laitat rastin siihen ja lähetät lomakkeen, henkilötietosi sekä lisäämäsi nuottijulkaisut poistuvat nuottiarkistosta. Poistosta ei anneta mitään varoituksia ja poiston jälkeen tulet ta-

kaisin rekisteröitymättömien etusivulle. Tämän jälkeen et voi enää sisäänkirjoittautua ilman rekisteröitymistä.

### 3.2.7 Nuotin lisäys arkistoon

Ollessasi rekisteröityneiden käyttäjien etusivulla (liite 3) tai muulla sivulla, josta on linkit rekisteröityneiden käyttäjien toimintoihin, voit valita linkin ”Nuotinisäys”. Tällöin pääset nuotinisäyssivulle (liite 4)

Sivulla on lomake, johon sinun tulee täyttää teoksen nimi, säveltäjän etu- ja sukunimi, aihepiiri (tai aihepiirit), koko (sivuina), URL-osoite kokonaisuena sekä lisätiedot.

URL-osoite ja teoksen nimi ovat pakollisia tietoja, mutta muidenkin tietojen täyttäminen on äärimmäisen suositeltavaa.

Säveltäjä-kohdan ylempi kenttä on säveltäjän etunimelle ja alempi sukunimelle. Aihepiirit -valikosta voit valita useammankin vaihtoehdon sillä lista on monivalittava. Jos mikään vaihtoehto ei ole sopiva, voit kirjoittaa oman uuden aihepiirin kenttään ”Uusi aihepiiri”, joka siis on monivalintalistalla. Voit myös valita valmiina olevia aihepiirejä ja sen lisäksi lisätä oman aihepiirin. Jos haluat liittää lisäämäsi nuottijulkaisun ainoastaan lisäämääsi aihepiiriin tarkista, että monivalintalistasta on valittu ”(ei valittu)”.

Lisätietoihin voit liittää kuvauksen teoksesta (enimmäispituus 4000 merkkiä). On suositeltavaa liittää lisätietoihin avainsanoja, joiden perusteella teos on helppo hakea kysynnän perusteella löytää. Muista, että mikään sivun tekstikentistä ei saa sisältää hipsuja.

Muista vielä olevasi vastuussa siitä, että teos on luvallisesti esillä. Nuottiarkistoon ei saa lisä tä linkkejä tekijänoikeuksia rikkoviin sivustoihin. Tällaiset nuottijulkaisut voi arkistonhoitaja halutessaan poistaa.

### 3.2.8 Nuottijulkaisun poisto arkistosta

Ollessasi rekisteröityneiden käyttäjien etusivulla (liite 3) tai muulla sivulla, josta on linkit rekisteröityneiden käyttäjien toimintoihin, voit valita linkin ”Nuotinpoisto”. Tällöin pääset nuotinpoistosivulle (liite 5)

Sivulla näkyvät lisäämäsi nuottijulkaisut taulukossa. Jokaisesta julkaisuista näkyvät säveltäjä, nuottijulkaisun nimi, URL-osoite ja lisäyspäivä. Valitse poistettavat julkaisut säveltäjien vasemmalla olevista rasteista ja paina nappia ”Poista valitut arkistosta”. Poistoon ei pyydetä varmistusta ja poistoa ei voi peruuttaa. Voit myös poistaa rastit eli poistaa valinnat painamalla nappia ”Tyhjennä”.

Poistettuasi nuottijulkaisut, ne häviävät listalta. Voit jatkaa sivun linkeistä.

Jos olet arkiston hoitaja, katso nuottijulkaisujen poiston ohjeet kohdasta 3.3.

### 3.2.9 Lopetus

Voit kirjoittautua ulos järjestelmästä valitsemalla rekisteröityneiden käyttäjien etusivulla, tai muulla sivulla jossa on linkki, vaihtoehdon ”Lopetus”. Tällöin siirryt aloitussivulle, siis rekisteröimättömien etusivulle ja käyttääksesi rekisteröityneiden toimintoja joudut uudestaan kirjoittautumaan sisälle järjestelmään.

## 3.3 Arkiston hoitajalle

Jos olet arkiston hoitaja, sinulle ei ohjelman käytössä ole muita eroja rekisteröityneiden käyttäjien käyttöön verrattuna paitsi nuottijulkaisujen poisto arkistosta. Sisäänkirjoittautuminen tapahtuu siis normaalisti, voit muuttaa henkilötietojasi normaalisti (paitsi käyttäjätunnusta) ja muutenkin muiden kuin nuotinpoiston osalta et näe ohjelmassa mitään erilaista.

### 3.3.1 Nuottijulkaisujen poisto

Ollessasi rekisteröityneiden käyttäjien etusivulla (liite 3 tai muulla sivulla, josta on linkit rekisteröityneiden käyttäjien toimintoihin, voit valita linkin ”Nuotinpoisto”. Tällöin pääset arkiston hoitajan nuotinpoistosivulle (liite D).

Sivulla näkyvät kaikki arkistossa sillä hetkellä olevat nuottijulkaisut taulukossa. Jokaisesta julkaisuista näkyvät säveltäjä, nuottijulkaisun nimi, URL-osoite ja lisäyspäivä. Lisäksi näkyy nuottijulkaisun lisänneen henkilön käyttäjätunnus. Teokset on järjestetty käyttätunnusten sekä säveltäjien ja teosten nimien mukaan aakkosjärjestykseen. URL-osoitteesta voit tarkistaa linkin toimivuuden normaalisti ja käyttäjätunnukset toimivat sivun sisäisinä linkkeinä käyttäjien tietoihin.

Käyttäjien tiedot on käyttäjien nimien mukaisessa aakkosjärjestyksessä ja jokaisesta käyttäjästä ovat taulukossa suku- ja etunimi, sähköpostiosoite (mikäli annettu) sekä käyttäjätunnus. Sähköpostiosoite toimii suorana linkkinä käyttäjän sähköpostiin.

Valitse poistettavat julkaisut säveltäjien vasemmalla olevista rasteista ja paina nappia ”Poista valitut arkistosta”. Poistoon ei pyydetä varmistusta ja poistoa ei voi peruuttaa. Voit myös poistaa rastit eli poistaa valinnat painamalla nappia ”Tyhjennä”.

Hyviä ja musiikillisia hetkiä Nuottiarkiston parissa!

## A Tehtäväkuvaus

Sävelteosten nuottiarkisto IS98JE2

Sävelteosten nuottiarkistoon tallennetaan tietoja yksittäisistä nuottidokumenteista. Kullakin nuottidokumentilla on tunnistetiedot ja lyhyt noin rivin mittainen selitys nuotin sisällöstä. Nuottidokumentit voivat sijaita hajautetusti Internetissä. Kunkin nuottidokumentin URL-osoite ja mikäli mahdollista myös kokotiedot tallennetaan arkistoon. Nuottidokumenttiin ja kokoelmiin voidaan liittää myös tekijätietoja (myös nuotin luovuttajasta (nimi, e-mail)). Nuottidokumentit luokitellaan aihepiireittäin. Nuottidokumentti voi kuulua useaan aihepiiriin. Nuottidokumenttiin voidaan liittää myös avainsanoja. Jokaiseen nuottidokumenttiin liittyy aikaleima, joka kertoo milloin nuotti on lisätty arkistoon. Arkiston pitäjä haluaa seurata arkiston käyttöä. Tätä varten liitetään jokaiseen nuottidokumenttiin laskuri.

Nuottiarkistoa voidaan käyttää eri tavoin. Käyttäjä voi valita jonkin aihepiirin ja selata siihen liittyviä nuotteja. Selaus pitäisi pystyä rajoittamaan myös uutuuksiin. Nuotteja voi hakea myös asiasanan ja tekijätietojen perusteella. Tällöinkin pitäisi pystyä hakemaan vain uutuuudet. Käyttäjällä pitäisi olla mahdollisuus liittää arkistoon uusia nuotteja.

Toimintoja (toteutettavat valitaan):

- a) Nuottidokumentin lisäys arkistoon
- b) Nuottidokumentin poisto arkistosta
- c) Nuottidokumenttien haku ja katselu
- d) Uutuusluettelot
- e) Yhteenvedoja nuottidokumenttien käytöstä

## B Lähdekoodit

```

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

/*=====
Servletti Nuottiarkiston etusivun generoimiseen.
Luokka kokoaa etusivun luokan Etusivutekstit tarjoamista palasista ja
käyttää luokan Html palveluja.

kirjoittanut: Ahti Syreeni
pvm:          23.4.2001
=====*/
public class Etusivu extends HttpServlet{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        /**Metodi generoi etusivun. Jos käyttäjä on kirjoittautunut sisään,
        generoidaan rekistereityneiden sivu, muuten normaali*/

        ServletOutputStream palaute;
        res.setContentType("text/html");
        palaute= res.getOutputStream();
        boolean rekisteroitunut=false;
        Cookie[] piparit=req.getCookies();

        //tutkitaan, onko rekisteroitynyt kayttaja
        for(int i=0;i<piparit.length;i++){
            if(piparit[i].getName().equals("nuottiarkisto")){
                if(!(piparit[i].getValue()==null))
                    rekisteroitunut=true;
            }
        }
        palaute.println(Etusivutekstit.aluste()); //sivun perusosat
        palaute.println(Etusivutekstit.otsikko());

        if(rekisteroitunut){ //generoidaan eri sivut rekistereityneille
            palaute.println(Etusivutekstit.rekEsittely());
            palaute.println(Etusivutekstit.rekLinkit());
        }
        else{
            palaute.println(Etusivutekstit.esittely());
        }
    }
}

```

```

//luodaan yhteys tietokantaan
Connection yhteys=null;
yhteys= Tietokanta.luoYhteys();
if (yhteys==null) {
    palaute.println(Virheilmoitukset.tkYhteysvirhe());
    return;
}

//haetaan aihepiirien nimet tietokannasta

Statement stmt = null;
ResultSet hakutulos = null;
String[] tulostaulukko;

try {
    stmt = yhteys.createStatement();
    hakutulos = stmt.executeQuery("SELECT DISTINCT nimi FROM "+
                                   "aihepiiri ORDER BY nimi");

    tulostaulukko=new String[hakutulos.getFetchSize()];

    for(int i=0; i<hakutulos.getFetchSize();i++)
        if(hakutulos.next())
            tulostaulukko[i]=hakutulos.getString(1);

    //generoidaan hakupalvelu kyselystä saaduilla aihepiireillä

    palaute.println(Etusivutekstit.hakulomake(tulostaulukko));
    if(!rekisteroitunut)
        palaute.println(Etusivutekstit.login());
    palaute.println(Etusivutekstit.ohjeet());
    palaute.println(Html.lopetus());
}catch (SQLException e){
    palaute.println(Virheilmoitukset.SQLVirhe(e.getMessage()));
}finally{
    try {
        if (hakutulos!=null) hakutulos.close();
        if (stmt!=null) stmt.close();
        Tietokanta.suljeYhteys(yhteys);
    } catch(SQLException e) {}
}
}
}

/*=====

```



Kirjastoluokka Nuottiarkiston etusivun tekstien tallentamiseen  
Kaikki tekstit ovat html-muodossa ja luokan metodit palauttavat  
tekstit html-muodossa. Joidenkin html-rakenteiden luomiseen käytetään  
myös Html-luokan palveluja.

kirjoittanut: Ahti Syreeni

pvm: 21.4.2001

=====\*/

```
public class Etusivutekstit{
    //*****Hakemistopolut kuville*****
    private static final String otsikko="Nuottiarkisto"; //sivun otsikko
    private static final String otsikkokuva=
        "http://db.cs.helsinki.fi/u/asyreeni/nuottiarkisto.jpg";
    private static final String tausta=
        "http://db.cs.helsinki.fi/u/asyreeni/tausta.jpg";
    //*****
    //*****sisäänkirjoittautuminen*****
    private static final String login="+
    /**lomake sisäänkirjoittautumista varten*/
    "<center>\n"+
    "<form method=\"post\" action=./Login>\n"+
    "  <table border=3>\n"+
    "    <tr>\n"+
    "      <td><label for=\"ktunnus\">K&auml;ytt&auml;j&auml;tunnus:\n"+
    "        </label> <input type=\"text\" name=\"ktunnus\" size=15\n"+
    "          maxlength=10></td>\n"+
    "      <td><label for=\"salasana\">Salasana:</label>\n"+
    "        <input type=\"password\" name=salasana size=15\n"+
    "          maxlength=10></td>\n"+
    "    <tr>\n"+
    "      <td>Ei k&auml;ytt&auml;j&auml;tunnusta? <a href=\""+
    "        \"Rekisterointilomake\">Rekister&ouml;idy.</a></td>\n"+
    "      <td><input type=\"submit\" value=\"L&auml;het&auml;\">\n"+
    "        <input type=\"reset\" value=\"Tyhjenn&auml;\"></td>\n"+
    "    </table>\n"+
    "</form>"+
    "</center>";
    //*****Hakupalvelulomake*****
    private static final String hakupalveluA="+
    /**Hakupalvelun osat ennen vaihtuvaa aihepiiri-kenttää */
    "<form method=\"post\" action=./Haku>\n"+
    "  <table border=1 style=\"width:100%;\">\n"+
    "    <col style=\"width:30%;\"><col style=\"width:70%;\">\n"+
    "  <tr>\n"+
```

```

"      <td><input type="checkbox\" name=vhakusana><label for=\n"+
"          \"hakusana\">Hakusana:</label></td>\n"+
"      <td><input type="text\" name=hakusana style=\n"+
"          \"width:100%;\"></td>\n"+
"  <tr>\n"+
"      <td><input type="checkbox\" name=vteosnimi><label for=\n"+
"          \"teosnimi\">Teoksen nimi:</label></td>\n"+
"      <td><input type="text\" name=teosnimi "+
"          \"style="width:100%;\">\n"></td>\n"+
"  <tr>\n"+
"      <td><input type="checkbox\" name=saveltaja><label for\n"+
"          =\"sukunimi\">S&auml;velt&auml;j&auml;:</label></td>\n"+
"      <td><label for="etunimi\">Etunimi:</label><input type=\n"+
"          \"text\" name=etunimi style="width:100%;\">\n"+
"          <label for="sukunimi\">Sukunimi:</label><input type=\n"+
"          \"text\" name=sukunimi style="width:100%;\"></td>\n"+
"  <tr>\n";

private static final String hakupalveluB=""+
/**hakupalvelulomakkeen loppuosa, aihepiiri-kentän jälkeen*/
"      <tr>\n"+
"      <td><input type="checkbox\" name=vvuosi><label for=\n"+
"          \"svuosi\">S&auml;vellysvuosi:</label></td>\n"+
"      <td><input type="text\" name=svuosi style="width:50%;\">\n"+
"          maxlength=4>\n"+
"          <select name=svuositarkkuus style="width:50%;\">\n"+
"              <option selected>tasan\n"+
"              <option>jalkeen\n"+
"              <option>ennen\n"+
"          </select></td>\n"+
"  <tr>\n"+
"      <td><input type="checkbox\" name=vuutuudet><label\n"+
"          for="uutuudet\">Vain uutuudet:</label></td>\n"+
"      <td>\n"+
"          <select name=uutuudet style="width:100%;\">\n"+
"              <option selected>1 kk\n"+
"              <option>2 kk\n"+
"              <option>3 kk\n"+
"              <option>4 kk\n"+
"              <option>5 kk\n"+
"              <option>6 kk\n"+
"              <option>7 kk\n"+
"              <option>8 kk\n"+
"              <option>9 kk\n"+
"              <option>10 kk\n"+

```

```

"          <option>11 kk\n"+
"          <option>12 kk\n"+
"        </select></td>\n"+
"      <tr>\n"+
"        <td><input type=\"submit\" value=\"L&auml;het&auml;\">\n"+
"          <input type=\"reset\" value=\"Tyhjenn&auml;\"></td>\n"+
"      </table>\n"+
"</form>";

//=====
//*****metodit*****

public static String aluste(){
    /**palauttaa sivun alun Html-määrittelyn otsikolla*/
    return Html.aluste("Nuottiarkisto",tausta);
}
//-----

public static String otsikko(){
    /**palauttaa etusivun otsikon html-muodossa, kuvaosoite voidaan myös
    jättää tyhjäksi alun määrittelyssä*/

    if(!(otsikkokuva==null))
        return Html.otsikkokuva(otsikko, otsikkokuva);
    else
        return Html.otsikko(otsikko,1);
}
//-----

public static String esittely(){
    /**palauttaa Nuottiarkiston yleisesittelyn*/
    return
        "<p>Nuottiarkisto on www-pohjainen nuottijulkaisujen tietojen\n"+
        "  s&auml;il&auml;mispaikka. Arkistoon ker&auml;t&auml;&auml;n "+
        "  Internetiss&auml; saatavilla\n"+
        "  olevien nuottijulkaisujen tietoja ja linkit niihin.\n"+
        "  Nuottiarkistosta voidaan hakea nuottijulkaisuja erilaisin\n"+
        "  hakuehdoin ja nuottijulkaisuja voidaan sek&auml; lis&auml;t"+
        "  &auml;\n"+
        "  ett&auml; poistaa. <em>Lis&auml;&auml;minen ja poistaminen"+
        "  edellytt&auml;t\n "+Html.linkki(
            "rekister&auml;itymist&auml;.",
            "Rekisterointilomake")+"</em>\n "+
        "  Jos olet rekister&auml;itynyt, voit kirjoittautua "+
        "  sis&auml;&auml;n hakulomakkeen alla olevalla pienell&auml; "+
        "  lomakkeella."+
        "</p>\n";
}

```

```

}
//-----
public static String rekEsittely(){
    /**palauttaa rekister&ouml;ityneiden käyttäjien etusivun esittelyn */
    return
        "<p><strong>Koska olet rekister&ouml;itynyt</strong>, voit valita"+
        " vapaasti seuraavista vaihtoehdoista. <em>Muista kuitenkin\n"+
        " olevasi vastuussa lis&auml;&auml;mist&auml;si teoksista."+
        " Nuottiarkistoon\n"+
        " ei saa lis&auml;t&auml; linkkej&auml; tekij&auml;noikeuksia"+
        " rikkoviin \n nuottijulkaisuihin.</em></p>\n";
}
//-----
public static String ohjeet(){
    /**palauttaa etusivun ohjeet */
    return
        "<h2>Ohjeita arkistosta hakuun</h2>\n"+
        " <p>Jokainen hakuvaihtoehto, joka halutaan hakuperusteeksi,\n"+
        " on merkitt&auml;v&auml;lomakkeen vasemmassa reunassa\n "+
        " olevalla rastilla. Useita hakuvaihtoehtoja voidaan\n "+
        " merkit&auml;kerralla hakuvaihtoehtoiksi, jolloin\n "+
        " haettavien nuottijulkaisujen on t&auml;ytett&auml;v&auml;\n "+
        " kaikki merkityt hakukriteerit. Mik&auml;li hakuperuste on\n "+
        " merkitty, mutta sille ei ole annettu mit&auml;&auml;n arvoja,\n "+
        " hakuperustetta ei huomioida haussa. T&auml;ytetty&auml;si\n"+
        " hakulomakkeen suorita haku painamalla lomakkeen alhaalla\n"+
        " olevaa \"L&auml;heta\"-nappia. Voit my&ouml;s\n "+
        " tyhj&auml;t&auml;lomakkeen \"Tyhjenn&auml;\"-napista.\n"+
        " </p>\n"+
        "<ul>\n"+
        " <li>Hakusana\n"+
        " <ul>\n"+
        " <li>Jos merkitset t&auml;m&auml;n hakuperusteen ja annat\n"+
        " jonkin sanan, nuottiarkistosta haetaan kaikki ne \n"+
        " nuottijulkaisut, joiden s&auml;velt&auml;j&auml;n\n"+
        " nimess&auml;, teoksen nimess&auml;, URL-osoitteessa, \n"+
        " aihepiireiss&auml; tai lis&auml;tiedoissa esiintyy\n"+
        " antamasi sana. Antamasi sanan kirjainkoolla ei ole \n"+
        " merkityst&auml; ja hakusanan ei tarvitse olla\n"+
        " kokonainen sana ja hakusana voi sijaita \n"+
        " nuottijulkaisun tiedoissa my&ouml;s sanan\n"+
        " keskell&auml;. Niinp&auml; esimerkiksi antamalla\n"+
        " hakusanaksi kirjain a, saadaan hakutulokseksi kaikki\n"+
        " ne teokset, joiden edell&auml;luetelluissa tiedoissa\n"+
        " a-kirjain esiintyy. Hakutuloksia voi siis rajoittaa\n"+

```

```

"         antamalla pitempi&auml; hakusanoja. <strong>Hakusana \n"+
"         ei saa sis&auml;lt&auml;&auml; hipsuja (')</strong></li>\n"+
"     </ul></li>\n"+
" <li>Teoksen nimi\n"+
"     <ul>\n"+
"         <li>Hakuehdolla haetaan kaikki ne teokset, joiden\n"+
"             nimen alussa esiintyy annettu nimi.<strong>Teoksen\n"+
"             nimi ei saa sis&auml;lt&auml;&auml; hipsuja (')\n"+
"                 </strong></li>\n, kirjainkoolla ei ole v&auml;li&auml;"+
"     </ul></li>\n"+
" <li>S&auml;velt&auml;j&auml;\n"+
"     <ul>\n"+
"         <li>Hakuehdolla haetaan kaikki ne teokset joiden\n"+
"             s&auml;velt&auml;j&auml;n nimess&auml; on annettu sana."+
"             S&auml;velt&auml;j&auml;n nimet\n"+
"             eiv&auml;t saa sis&auml;lt&auml;&auml; hipsuja"+
"             (').<strong>\n Toinen nimist&auml; voidaan olla tyhj&auml;"+
"             Huom! Joissakin selaimissa ei n&auml;y tekstej&auml;"+
"             jotka ilmoittavat, ett&auml; ylempi nimikentt&auml; on\n"+
"             s&auml;velt&auml;j&auml;n etunimi ja alempi kentt&auml;\n"+
"             s&auml;velt&auml;j&auml;n sukunimi.</strong></li>\n"+
"     </ul></li>\n"+
" <li>Aihepiiri\n"+
"     <ul>\n"+
"         <li>Hakuehdolla haetaan kaikki ne teokset, jotka\n"+
"             kuuluvat valittuun aihepiiriin. Sama teos voi \n"+
"             kuulua useampaankin aihepiiriin.</li>\n"+
"     </ul></li>\n"+
" <li>S&auml;vellysvuosi\n"+
"     <ul>\n"+
"         <li>Hakuehdolla haetaan kaikki ne teokset, jotka\n"+
"             on s&auml;velletty annettuna vuonna, sit&auml; ennen,\n"+
"             tai sen j&auml;lkeen. Vuosiluku annetaan aina \n"+
"             t&auml;ydellisen&auml; (esim.1995).</li>\n"+
"     </ul></li>\n"+
" <li>Vain uutuudet\n"+
"     <ul>\n"+
"         <li>Hakuehdolla haetaan kaikki ne teokset, jotka\n"+
"             on lis&auml;tty enint&auml;&auml;n valittu \n"+
"             kuukausim&auml;r&auml; sitten. \n"+
"     </ul></li>\n"+
"</ul>\n";
}
//-----

```

```

public static String login(){
    /**palauttaa sisaankirjoittautumisen lomakkeen*/
    return login;
}
//-----
public static String hakulomake(String[] aihepiirit){
    /**generoi html-hakulomakkeen annetuilla aihepiireilla*/

    String lomake;
    lomake=hakupalveluA;          //ensin lomakkeen ekat pysyvät osat
                                //sitten alustus aihepiirivalikolle
    lomake+="      <td><input type=\"checkbox\" name=vaihepiiri>"+
              "<label\n"+
              "      for=\"aihepiiri\">Aihepiiri:</label></td>"+
              "      <td><select name=aihepiiri style=\"width:100%;\">\n";

    if(aihepiirit.length>0){    //sitten annetut aihepiirit valikkoon
        if(!(aihepiirit[0]==null))
            lomake+="      <option selected>"+aihepiirit[0]+" \n";
        if(aihepiirit.length>1)
            for(int i=1;i<aihepiirit.length;i++)
                if(!(aihepiirit[i]==null))
                    lomake+="      <option>"+aihepiirit[i]+" \n";
        }
    lomake+="      </select>\n";

    lomake+=hakupalveluB;      //lomakkeen pysyvät loppuosat mukaan

    return lomake;
}
//-----
public static String rekLinkit(){
    /**palauttaa linkit rekisteröityneiden käyttäjien toimintoihin*/
    return
        Html.linkki("Nuotinlis&auml;ys", "Nuottilomake")+
        "--\n"+
        Html.linkki("Nuotinpoisto", "Nuotinpoistolomake")+
        "--\n"+
        Html.linkki("Henkil&ouml;tietojen muuttaminen",
            "Rekisterointilomake")+ "--\n"+
        Html.linkki("Lopetus",
            "Logout")+ "\n";
}
}

```

```

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Vector;

/*=====
Kirjastoluokka Nuottiarkiston haun suorittamiseen.
Luokka kokoaa hakusivun luokan Hakutulostekstit tarjoamista palasista ja
käyttää luokan Html palveluja.

kirjoittanut: Ahti Syreeni
pvm:          23.4.2001
=====*/

public class Haku extends HttpServlet{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        /**Suorittaa nuotistohaun annetuilla perusteilla*/
        ServletOutputStream palaute;
        res.setContentType("text/html");
        palaute= res.getOutputStream();
        boolean rekisteroitunut=false;
        Cookie[] piparit=req.getCookies();

        //tutkitaan, onko rekisteroitynyt kayttaja
        for(int i=0;i<piparit.length;i++){
            if(piparit[i].getName().equals("nuottiarkisto")){
                if(!(piparit[i].getValue()==null))
                    rekisteroitunut=true;
            }
        }

        //laaditaan annetuista parametreista SQL-kysely
        String kysely="";
        String parametri=null;

        //lisätään hakusanaehto kyselyyn, jos valittu
        if(!(req.getParameter("vhakusana")==null)){ //turhaan ruksittu??
            if(!(req.getParameter("hakusana").equals(""))){
                parametri=req.getParameter("hakusana");
                //etsitään hakusanaa kaikista sarakkeista
                kysely+="AND ((UPPER(sEtunimi) like '%" +parametri.toUpperCase()+
                    "') OR "+
                    " (UPPER(sSukunimi) like '%" +parametri.toUpperCase()+
                    "') OR "+
                    " (UPPER(tNimi) like '%" +parametri.toUpperCase()+

```

```

                                "%') OR "+
"      (UPPER(URLosoite) like '%" + parametri.toUpperCase() +
                                "%') OR "+
"      (UPPER(lisatiedot) like '%" + parametri.toUpperCase() +
                                "%') OR "+
"      (tunnistenumero IN "+
"      (select teosnumero from aiheyhteys, aihepiiri "+
"      where "+
"      aiheviite=aihenumero AND UPPER(nimi) like "+
"      '%" + parametri.toUpperCase() + "%'))");
}
}
//lisätään teoksen nimi hakuehdoksi, jos valittu
if(!(req.getParameter("vteosnimi")==null)){
    if(!(req.getParameter("teosnimi").equals(""))){
        parametri=req.getParameter("teosnimi");
        kysely+= " AND (UPPER(tNimi) like '%" + parametri.toUpperCase() + "%')";
    }
}
//lisätään teoksen säveltäjä hakuehdoksi, jos valittu
if(!(req.getParameter("saveltaja")==null)){
    if(!(req.getParameter("etunimi").equals(""))){
        parametri=req.getParameter("etunimi");
        kysely+= " AND (UPPER(sEtunimi) like '%" + parametri.toUpperCase() +
                                "%')";
    }
}
if(!(req.getParameter("sukunimi").equals(""))){
    parametri=req.getParameter("sukunimi");
    kysely+= " AND (UPPER(sSukunimi) like '%" + parametri.toUpperCase() +
                                "%')";
}
}
if(!(req.getParameter("vvuosi")==null)){
    //sävellysvuosi, tasan, ennen vai jälkeen?
    if(!(req.getParameter("svuosi").equals(""))){
        //tarkistetaan numeroarvon kelpoisuus
        try{
            Integer.parseInt(req.getParameter("svuosi"));
        }catch(Exception e){
            palaute.println(Virheilmoitukset.eiNumero());
            return;
        }
    }
    if(req.getParameter("svuositarkkuus").equals("tasan"))
        parametri="";
}

```



```

else
    if(req.getParameter("svuositarkkuus").equals("jalkeen"))
        parametri=">";
else
    if(req.getParameter("svuositarkkuus").equals("ennen"))
        parametri="<";
else{
    palaute.println(Html.aluste("KoodiVirhe"));
    palaute.println(Html.lopetus());
}
kysely+= " AND (savellysvuosi "+parametri;
parametri= req.getParameter("svuosi");
kysely+=" "+parametri+"");
}
}
if(!(req.getParameter("vaihepiiri")==null)){
    parametri= req.getParameter("aihepiiri");
    kysely+=" AND (tunnistenumero IN "+
        "(select teosnumero from aiheyhteys, aihepiiri "+
        " where aiheviite=aihenumero AND nimi like "+
        "'%"+parametri+''))";
}
if(!(req.getParameter("vuutuudet")==null)){
    try{
        parametri=req.getParameter("uutuudet").substring(0,1);
        parametri.trim();
        int paiva= Integer.parseInt(parametri) * 30; //muunnetaan valittu
                                                    //kuukausi päiväksi
        kysely+= " AND (lisayspaiva > (sysdate-"+paiva+"))";
    }catch(Exception e){
        palaute.println(Virheilmoitukset.SQLVirhe(e.getMessage()));
        return;
    }
}
//jos ei mitään valittu, ei kannata tehdä tyhjää kyselyä
if(kysely.equals("")){
    res.sendRedirect("Etusivu");
    return;
}
//liitetään kyselyyn varsinaiset SQL-elementit
kysely="select distinct sSukunimi, sEtunimi, tNimi, savellysvuosi,"+
    " koko, URLosoite, lisatiedot, lisayspaiva from teos, "+
    "saveltaja where (saveltaja=saveltajanumero) "+kysely;

```

```

kysely=kysely+" ORDER BY sSukunimi, sEtunimi, tNimi";

//luodaan yhteys tietokantaan ja suoritetaan saatu kysely
Connection yhteys=null;
yhteys= Tietokanta.luoYhteys();
if (yhteys==null) {
    palaute.println(Virheilmoitukset.tkYhteysvirhe());
    return;
}
Statement stmt = null;
ResultSet hakutulos = null;
Vector tulosvektori=new Vector(); //lisätiedot talteen
try {
    stmt = yhteys.createStatement();
    hakutulos = stmt.executeQuery(kysely);

    palaute.println(Hakutulostekstit.alkutekstit());
    palaute.println(Hakutulostekstit.linkit(rekisteroitunut));
    palaute.println(Hakutulostekstit.taulukko(true));
    palaute.println(Hakutulostekstit.sarakenimet());

    String apu;
    int laskuri=0;
    while(hakutulos.next()){
        //tulostetaan hakutulostaulukon rivit
        palaute.println(Hakutulostekstit.tulosrivi(
                                hakutulos.getString(1),
                                hakutulos.getString(2),
                                hakutulos.getString(3),
                                hakutulos.getString(4),
                                hakutulos.getString(5),
                                hakutulos.getString(6),
                                laskuri));

        //säästetään tiedot lisätieto-osioita varten
        //ensin estetään null-nimien pääsy tulosteeseen
        //pakko tehdä vaikeasti, jotta ei tule NullPointerException-poikkeusta
        apu=hakutulos.getString(1);
        if(apu==null)
            apu="-";
        else
            if(apu.equals("null"))
                apu="-";
        if(!(hakutulos.getString(2)==null)){
            if(hakutulos.getString(2).equals("null"))
                apu+=", ";

```

```

        else
            apu+=" "+hakutulos.getString(2);
    }
    tulosvektori.addElement(apu+" "+hakutulos.getString(3));
    tulosvektori.addElement(hakutulos.getString(7));
    tulosvektori.addElement(hakutulos.getString(8));
    laskuri++;
}
palaute.println(" "+Html.taulukkolopetus());
palaute.println(Hakutulostekstit.linkit(rekisteroitunut));

laskuri=0;
for(int i=0;i+2<tulosvektori.size();i+=3){
    //tulostetaan lisätiedot
    palaute.println(Hakutulostekstit.lisätiedot(
        (String)tulosvektori.elementAt(i),
        (String)tulosvektori.elementAt(i+1),
        (String)tulosvektori.elementAt(i+2),
        laskuri));
    laskuri++;
}
palaute.println(Hakutulostekstit.linkit(rekisteroitunut));
palaute.println(Html.lopetus());
}catch (SQLException e){
    //hakuehdossa virheitä, koska väärä merkkejä (=hipsuja) annettu
    palaute.println(Virheilmoitukset.kielletytMerkit());
}finally{
    try {
        if (hakutulos!=null) hakutulos.close();
        if (stmt!=null) stmt.close();
        Tietokanta.suljeYhteys(yhteys);
    } catch(SQLException e) {}
}
}
}

/*=====
Kirjastoluokka Nuottiarkiston haun tuloksien näyttämiseen ja tyyliin.
Luokka käyttää luokan Html palveluja.

kirjoittanut: Ahti Syreeni
pvm:          24.4.2001
=====*/
public class Hakutulostekstit{

```

```

//*****kuvien hakemistopolut ja sivun yleisasetelma*****
public static final String otsikko="Nuottihaun tulokset";
public static final String tausta=
    "http://db.cs.helsinki.fi/u/asyreeni/tausta.jpg";
public static final String taulukkotyylisi="style=\"width:100%\"";
public static final int taulukkoreunus=1;
//*****
public static String alkutekstit(){
    /**palauttaa hakutulossivun alkutekstit*/
    return Html.aluste("Nuottiarkisto-hakutulokset",tausta)+
        Html.otsikko(otsikko,1);
}
//-----
public static String taulukko(boolean alku){
    /**palauttaa hakutulostaulukon alusteen*/
    if(alku)
        return Html.taulukkoaluste(taulukkoreunus,taulukotyylisi);
    else
        return Html.taulukkolopetus();
}
//-----
public static String sarakenimet(){
    /**palauttaa hakutulostaulukon sarakenimet*/
    return " <tr>\n"+
        " <th>S&auml;velt&auml;j&auml;</th>\n"+
        " <th>Teoksen nimi</th>\n"+
        " <th>S&auml;vellysvuosi</th>\n"+
        " <th>Koko</th>\n"+
        " <th>URL-osoite</th>\n"+
        " </tr>\n";
}
//-----
public static String lisayspvm(String pvm){
    /**palauttaa lisäysmerkinnän*/
    return " <em>Lis&auml;tty: "+pvm+"</em>";
}
//-----
public static String linkit(boolean rekisteroitunut){
    /**palauttaa linkit muille sivuille*/
    if(rekisteroitunut)
        return
            "<p>"+Html.linkki("Etusivu ", "Etusivu")+ "--"+
            Html.linkki("Nuotinlisays", "Nuottilomake")+ "--\n " +
            Html.linkki("Nuotinpoisto", "Nuotinpoistolomake")+ "--"+
            Html.linkki("Lopetus", "Logout")+ "\n"+

```

```

        "</p>";
    else
        return
            "<p>" + Html.linkki("Etusivu ", "Etusivu") + "</p>";
    }
}
//-----
public static String tulorivi(String sukunimi, String etunimi,
                              String teos,    String svuosi,
                              String koko,    String url,
                              int    hakunro){
    /**palauttaa taulukkorivin annetuilla parametreilla*/

    //ei päästetä tietokannan null-arvoa näkyviin
    String nimi;
    if(sukunimi==null)
        sukunimi="-";
    else //vältetään NullPointerException
        if(sukunimi.equals("null"))
            sukunimi="-";
    if(etunimi==null)
        etunimi="-";
    else
        if(etunimi.equals("null"))
            etunimi="-";
    if( svuosi==null)
        svuosi="-";
    else
        if(svuosi.equals("null"))
            svuosi="-";
    if(koko==null)
        koko="-";
    else
        if(koko.equals("null"))
            koko="-";

    nimi=sukunimi+" "+etunimi;
    return
        " <tr>\n"+
        " <td>" + nimi + "</td>\n"+
        " <td>" + Html.linkki(teos, "#" + hakunro) + "</td>\n"+
        " <td>" + svuosi + "</td>\n"+
        " <td>" + koko + "</td>\n"+
        " <td>" + Html.linkki(url, url) + "</td>\n"+
        " </tr>\n";
}

```

```
//-----
public static String lisätiedot(String otsikko, String tieto,
                               String pvm,    int   nro){
    /**palauttaa lisätiedot html-muodossa */
    if(tieto==null)
        tieto="";
    else
        if(tieto.equals("null"))
            tieto="";

    return
        "<h3>"+Html.nimea(otsikko, ""+nro)+"</h3>\n"+
        "<p>\n"+tieto+"<br>\n<em>Lis&auml;tty: "+pvm+"</em>\n</p>";
}
}
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
/*=====
Servletti Nuottiarkistoon sisäänkirjoittautumiseen

kirjoittanut: Ahti Syreeni
pvm:          23.4.2001
=====*/
public class Login extends HttpServlet{
    /*******
    private static final String piparinimi="nuottiarkisto";
    /*******
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        /**suorittaa sisäänkirjoittautumisen*/
        ServletOutputStream palaute;
        res.setContentType("text/html");
        palaute= res.getOutputStream();
        String ktunnus=null;

        //luodaan yhteys tietokantaan
        Connection yhteys=null;
        yhteys= Tietokanta.luoYhteys();
        if (yhteys==null) {
            palaute.println(Virheilmoitukset.tkYhteysvirhe());
            return;
        }
    }
}
```

```

}
//haetaan käyttäjän salasana
Statement stmt = null;
ResultSet hakutulos = null;
String[] tulostaulukko;

try {
    stmt = yhteys.createStatement();
    hakutulos = stmt.executeQuery("SELECT DISTINCT salasana FROM"+
        " kayttajatiedot WHERE"+
        " kayttajatunnus='"+
        req.getParameter("ktunnus")
        +"'");

    if(hakutulos.next()){
        if(hakutulos.getString("salasana").equals(
            req.getParameter("salasana")) || (hakutulos.getString(
                "salasana").equals("null"))){
            ktunnus=req.getParameter("ktunnus");
            try{
                //jos salasana oli oikein, luodaan käyttäjälle
                //tunnistepipari
                Cookie tunnistepipari=new Cookie(piparinimi,ktunnus);
                tunnistepipari.setMaxAge(-1); //pipari häviää suljettaessa
                //selain

                res.addCookie(tunnistepipari);
                res.sendRedirect("Etusivu");
            }catch(Exception e){
                palaute.println(Virheilmoitukset.eiPipareita());
                return;
            }
        }
        else
            palaute.println(Virheilmoitukset.login());
    }
    else
        palaute.println(Virheilmoitukset.login());
}catch (SQLException e){
    //salasana tai käyttäjätunnus sisälsi kiellettyjä merkkejä
    palaute.println(Virheilmoitukset.kielletytMerkit());
}finally{
    try {
        if (hakutulos!=null) hakutulos.close();
        if (stmt!=null) stmt.close();
        Tietokanta.suljeYhteys(yhteys);
    } catch(SQLException e) {}
}

```

```

    }
  }
}

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

/*=====
Servletti Nuottiarkistoon sisäänkirjoittautumiseen

kirjoittanut: Ahti Syreeni
pvm:          3.5.2001
=====*/

public class Logout extends HttpServlet{
  /**=====
  private static final String piparinimi="nuottiarkisto";
  /**=====
  public void service(HttpServletRequest req, HttpServletResponse res)
      throws ServletException, IOException{

    /**suorittaa uloskirjoittautumisen*/
    ServletOutputStream palaute;
    res.setContentType("text/html");
    palaute= res.getOutputStream();

    try{
      //tuhotaan pipari kirjoittamalla päälle
      Cookie tunnistepipari=new Cookie(piparinimi,null);
      tunnistepipari.setMaxAge(0);
      res.addCookie(tunnistepipari);
      res.sendRedirect("Etusivu");
    }catch(Exception e){}
  }
}

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

/*=====
Servletti Nuottiarkiston rekisteröitymislomakkeen generoimiseen.

```



kirjoittanut: Ahti Syreeni

pvm: 26.4.2001

```

=====*/
public class Rekisterointilomake extends HttpServlet{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        /**generoi rekisteröitymislomakkeen*/

        ServletOutputStream palaute;
        res.setContentType("text/html");
        palaute= res.getOutputStream();
        String ktunnus=null;
        Cookie[] piparit=req.getCookies();

        for(int i=0;i<piparit.length;i++){
            if(piparit[i].getName().equals("nuottiarkisto")){
                if(!(piparit[i].getValue()==null))
                    ktunnus=piparit[i].getValue();
            }
        }

        if(ktunnus==null){ //uusi käyttäjä, generoidaan uusi lomake
            palaute.println(Rekisterointitekstit.aluste(false));
            palaute.println(Rekisterointitekstit.ohjeet(false));
            palaute.println(Rekisterointitekstit.reklomake());
            palaute.println(Rekisterointitekstit.linkit(false));
            palaute.println(Html.lopetus());
            return;
        }

        //kysymyksessä tietojen muuttaminen ja haetaan tämänhetkiset
        //tiedot tietokannasta

        Connection yhteys=null;
        yhteys= Tietokanta.luoYhteys();

        if (yhteys==null) {
            palaute.println(Virheilmoitukset.tkYhteysvirhe());
            return;
        }

        String kysely="SELECT kEtunimi, kSukunimi, sahkopostiosoite, "+
            "salasana FROM kayttajatiedot WHERE kayttajatunnus="+
            "'"+ktunnus+"'";

        Statement stmt = null;

```

```

ResultSet hakutulos = null;
try {
    stmt = yhteys.createStatement();
    hakutulos = stmt.executeQuery(kysely);
    if(hakutulos.next()){
        palaute.println(Rekisterointitekstit.aluste(true));
        palaute.println(Rekisterointitekstit.ohjeet(true));
        palaute.println(Rekisterointitekstit.muutoslomake(
            hakutulos.getString(1),
            hakutulos.getString(2),
            hakutulos.getString(3),
            ktunnus,
            hakutulos.getString(4)));
        palaute.println(Rekisterointitekstit.linkit(true));
        palaute.println(Html.lopetus());
    }
    else
        res.sendRedirect("Etusivu");

} catch (SQLException e){
    palaute.println(Virheilmoitukset.SQLVirhe(e.getMessage()));
}finally{
    try {
        if (hakutulos!=null) hakutulos.close();
        if (stmt!=null) stmt.close();
        Tietokanta.suljeYhteys(yhteys);
    } catch(SQLException e) {}
}
}
}
}

```

```

/*=====
Kirjastoluokka Nuottiarkiston rekisteröintilomakkeen tekstien
tallentamiseen. Kaikki tekstit ovat html-muodossa ja luokan metodit
palauttavat tekstit html-muodossa.

```

kirjoittanut: Ahti Syreeni

pvm: 26.4.2001

```

=====*/
public class Rekisterointitekstit{
    //*****
    private static final String tausta=
        "http://db.cs.helsinki.fi/u/asyreeni/tausta.jpg";
    //*****

```

```

public static String aluste(boolean muutoslomake){
    /**palauttaa sivun html-alusteen*/
    if(!muutoslomake){
        return Html.aluste("Nuottiarkisto -Rekister&ouml;tyminen",tausta)+
            Html.otsikko("Rekister&ouml;itymislomake",1);
    }
    else{
        return Html.aluste("Nuottiarkisto -Henkil&ouml;tietojen "+
            "muuttaminen",tausta)+
            Html.otsikko("Henkil&ouml;tietojen muuttaminen",
                1);
    }
}
}
//-----
public static String ohjeet(boolean muutoslomake){
    /**palauttaa ohjeet lomakkeen täyttämiseen*/
    if(muutoslomake)
        return " <p>\n"+
            "    Voit muuttaa kaikkia muita tietojasi, paitsi\n "+
            "    k&auml;ytt&auml;j&auml;tunnustasi\n.Voit my&ouml;s\n"+
            "    poistaa henkil&ouml;tietosi"+
            "    valitsemalla henkil&ouml;tietojen poiston\n. Samalla\n"+
            "    kuitenkin poistuvat my&ouml;s lis&auml;&auml;m&auml;si"+
            "    teokset Nuottiarkistosta.\n"+
            " </p>\n";
    if(!muutoslomake)
        return " <p>\n"+
            "    Ole hyv&auml; ja anna henkil&ouml;tietosi alla\n"+
            "    oleviin kenttiin. K&auml;ytt&auml;j&auml;tunnus ja\n"+
            "    sukunimi ovat pakollisia.\n"+
            " </p>\n"+
            " <p>\n"+
            "    <strong>K&auml;ytt&auml;j&auml;tunnus on\n"+
            "    enimill&auml;&auml;n kymmenen kirjaimen mittainen.\n"+
            "    Salasana annetaan molempiin\n"+
            "    salasanakenttiin ja niiden on oltava\n"+
            "    t&auml;sm&auml;lleen samat. \n Ota huomioon,\n"+
            "    ett&auml; arkistonhoitaja n&auml;kee salasanasi.\n"+
            "    Mitk&auml;&auml;n sivun kent&auml;t eiv&auml;t saa \n"+
            "    sis&auml;lt&auml; hipsuja. "+
            " </strong> L&auml;hetetty&auml;si lomakkeen, p&auml;"+
            "    &auml;set rekister&ouml;ityneiden etusivulle."+
            " </p>\n";
    return null;
}

```

```

//-----
public static String reklomake(){
    /**palauttaa rekisteröitymislomakkeen*/
    return
        "<form method=\"post\" action=Rekisterointi>\n"+
        "<table style=\"width:100%;\">"+
        "  <col style=\"width:30%;\"><col style=\"width:70%;\">\n"+
        "  <tr>\n"+
        "    <td><label for=\"etunimi\">Etunimi:</label></td>\n"+
        "    <td><input type=\"text\" name=etunimi maxlength=20 style=\n"+
        "      \n\"width:100%;\"></td>\n"+
        "  </tr>\n"+
        "  <tr>\n"+
        "    <td><em><label for=\"sukunimi\">Sukunimi:</label></em></td>\n"+
        "    <td><input type=\"text\" name=sukunimi maxlength=20\n"+
        "      style=\"width:100%;\"></td>\n"+
        "  </tr>\n"+
        "  <tr>\n"+
        "    <td><label for=\"sposti\">S&auml;hk&ouml;posti:</label></td>\n"+
        "    <td><input type=\"text\" name=sposti maxlength=50\n"+
        "      style=\"width:100%;\"></td>\n"+
        "  </tr>\n"+
        "  <tr>\n"+
        "    <td><em><label for=\"ktunnus\">K&auml;ytt&auml;j&auml;tunnus:\n"+
        "      </label></em></td>\n"+
        "    <td><input type=\"text\" name=ktunnus maxlength=10\n"+
        "      style=\"width:100%;\"></td>\n"+
        "  </tr>\n"+
        "  <tr>\n"+
        "    <td><label for=\"salasana1\">Salasana:</label></td>\n"+
        "    <td><input type=\"password\" name=salasana1 maxlength=10\n"+
        "      style=\"width:100%;\"></td>\n"+
        "  </tr>\n"+
        "  <tr>\n"+
        "    <td><label for=\"salasana2\">Salasana:</label></td>\n"+
        "    <td><input type=\"password\" name=salasana2 maxlength=10\n"+
        "      style=\"width:100%;\"></td>\n"+
        "  </tr>\n"+
        "  <tr>\n"+
        "    <td><input type=\"submit\" value=\"L&auml;het&auml;\">\n"+
        "    <input type=\"reset\" value=\"Tyhjenn&auml;\"></td> \n"+
        "  </tr> \n "+
        "</table>\n";
}
//-----

```

```

public static String muutoslomake(String eNimi, String sNimi,
                                String email, String kTunnus,
                                String sSana){
    /**palauttaa henkilötietojen muutoslomakkeen esitötettyinä
       annetuilla parametereilla*/

    //poistetaan null-arvot (ja vältetään null-poikkeus...)
    if(eNimi==null)
        eNimi="";
    else
        if(eNimi.equals("null"))
            eNimi="";
    if(email==null)
        email="";
    else
        if(email.equals("null"))
            email="";
    if(sSana==null)
        sSana="";
    else
        if(sSana.equals("null"))
            sSana="";
    return
        ("<form method=\"post\" action=Rekisterointi>\n"+
         "<table style=\"width:100%;\">\n"+
         "  <col style=\"width:30%;\"><col style=\"width:70%;\">\n"+
         "  <tr>\n"+
         "    <td><label for=\"etunimi\">Etunimi:</label></td>\n"+
         "    <td><input type=\"text\" name=etunimi maxlength=20\"+
         "      value=\""+eNimi+"\" style=\"width:100%;\"></td>\n"+
         "  </tr>\n"+
         "  <tr>\n"+
         "    <td><em><label for=\"sukunimi\">Sukunimi:</label></em></td>\n"+
         "    <td><input type=\"text\" name=sukunimi maxlength=20\n"+
         "      value=\""+sNimi+"\" style=\"width:100%;\"></td>\n"+
         "  </tr>\n"+
         "  <tr>\n"+
         "    <td><label for=\"sposti\">S&auml;hk&ouml;posti:</label>\n"+
         "      </td>\n"+
         "    <td><input type=\"text\" name=sposti maxlength=50\n"+
         "      value=\""+email+"\" style=\"width:100%;\"></td>\n"+
         "  </tr>\n"+
         "  <tr>\n"+
         "    <td><em><label for=\"ktunnus\">K&auml;ytt&auml;j&auml;"+
         "      tunnus:\n"+

```

```

"         </label></em></td>\n"+
"     <td>"+kTunnus+"</td>\n"+
" </tr>\n"+
" <tr>\n"+
"     <td><label for=\"salasana1\">Salasana:</label></td>\n"+
"     <td><input type=\"password\" name=salasana1 maxlength=10\n"+
"         value=\""+sSana+"\" style=\"width:100%;\"></td>\n"+
" </tr>\n"+
" <tr>\n"+
"     <td><label for=\"salasana2\">Salasana:</label></td>\n"+
"     <td><input type=\"password\" name=salasana2 maxlength=10\n"+
"         value=\""+sSana+"\" style=\"width:100%;\"></td>\n"+
" </tr>\n"+
" <tr>\n"+
"     <td><input type=\"checkbox\" name=poisto>Henkil&ouml;tietojen"+
"         poisto\n"+
"     <td><input type=\"submit\" value=\"L&auml;het&auml;\">\n"+
"     <input type=\"reset\" value=\"Tyhjenn&auml;\"></td>\n "+
" </tr> \n "+
"</table>\n");
}
//-----
public static String linkit(boolean muutoslomake){
    /**palauttaa sivun linkit*/
    String palaute=" <p>\n"+      "+Html.linkki("Etusivu ", "Etusivu");
    if(muutoslomake)
        palaute+="\n--"+Html.linkki("Nuotinlisays",
            "Nuottilomake")+ "\n"+
            "--"+Html.linkki("Nuotinpoistolomake",
            "Nuotinpoistolomake")+ "\n"+
            "--"+Html.linkki("Lopetus", "Logout")+ "\n";
    palaute+=" </p>\n";
    return palaute;
}
}

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

/*=====
Servletti Nuottiarkiston rekisteröintilomakkeen käsittelemiseen.

```

kirjoittanut: Ahti Syreeni

pvm: 23.4.2001

```

=====*/
public class Rekisterointi extends HttpServlet{
    /*******
    //nimi, josta tunnustetaan nuottiarkiston tekemä tunnustepipari
    private static final String piparinimi="nuottiarkisto";
    /*******
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
    /**lisää rekisteröityneen käyttäjän tiedot tietokantaan*/

    ServletOutputStream palaute;
    res.setContentType("text/html");
    palaute= res.getOutputStream();
    String ktunnus=null;
    Cookie[] piparit=req.getCookies();

    //onko salasanan varmistus oikein...
    if(!(req.getParameter("salasana1").equals(req.getParameter(
        "salasana2")))){
        palaute.println(Virheilmoitukset.salasanavirhe());
        return;
    }

    //onko sukunimi varmasti annettu?
    if(req.getParameter("sukunimi").equals("")){
        palaute.println(Virheilmoitukset.sukunimi());
        return;
    }

    //onko kysymyksessä uusi käyttäjä?
    for(int i=0;i<piparit.length;i++){
        if(piparit[i].getName().equals("nuottiarkisto")){
            if(!(piparit[i].getValue()==null))
                ktunnus=piparit[i].getValue();
        }
    }

    String kysely=null;
    if(ktunnus==null){ //tehdään uusi rekisteröinti
        if(req.getParameter("ktunnus").equals("")){
            //käyttäjätunnusta ei annettu
            palaute.println(Virheilmoitukset.ktunnus());
            return;

```

```

}
//leivotaan pipari ja katsotaan onnistuuko
try{
    Cookie tunnistepipari=new Cookie(piparinimi,
                                     req.getParameter("ktunnus"));
    tunnistepipari.setMaxAge(-1); //pipari häviää suljettaessa
                                     //selain
    res.addCookie(tunnistepipari);
}catch(Exception e){
    //käyttäjätunnus sisältää kiellettyjä merkkejä
    palaute.println(Virheilmoitukset.ktunnusLaiton());
    return;
}
kysely="INSERT INTO kayttajatiedot VALUES("+
        "'"+req.getParameter("ktunnus")+"'"+
        ", '"+req.getParameter("etunimi")+"'"+
        ", '"+req.getParameter("sukunimi")+"'"+
        ", '"+req.getParameter("sposti")+"'"+
        ", '"+req.getParameter("salasana")+"'"+
        ", '");
}
else{ //muutetaan jo tietokannassa olevia tietoja
    if(req.getParameter("poisto")==null)
        kysely="UPDATE kayttajatiedot SET "+
                "kEtunimi='"+req.getParameter("etunimi")+"', "+
                "kSukunimi='"+req.getParameter("sukunimi")+"', "+
                "sahkopostiosoite='"+req.getParameter("sposti")+"', "+
                "salasana='"+req.getParameter("salasana1")+"' "+
                "WHERE kayttajatunnus='"+ktunnus+"'";
    else
        kysely="DELETE from kayttajatiedot WHERE kayttajatunnus='"+
                ktunnus+"'";
}
Connection yhteys=null;
yhteys= Tietokanta.luoYhteys();

if (yhteys==null) {
    res.sendRedirect("Logout");
    palaute.println(Virheilmoitukset.tkYhteysvirhe());
    return;
}

Statement stmt = null;
int onnistuminen=0;
try {

```



```

stmt = yhteys.createStatement();
onnistuminen= stmt.executeUpdate(kysely);
if(onnistuminen==0)
    palaute.println(Virheilmoitukset.SQLVirhe("Lisays ei onnistunut"));
else{
    if(req.getParameter("poisto")==null)
        res.sendRedirect("Etusivu");
    else
        res.sendRedirect("Logout");
}
}catch (SQLException e){
    try{
        //jo annettu tunnistepipari on tuhottava
        Cookie tunnistepipari2=new Cookie(piparinimi,null);
        tunnistepipari2.setMaxAge(0);
        res.addCookie(tunnistepipari2);
    }catch(Exception a){}
    palaute.println(Virheilmoitukset.kielletytMerkit());
}finally{
    try {
        if (stmt!=null) stmt.close();
        Tietokanta.suljeYhteys(yhteys);
    } catch(SQLException e) {}
}
}
}
}

```

```

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

```

```

/*=====

```

```

Servletti Nuottiarkiston nuotinnlisäyslomakkeen generoimiseen.

```

```

kirjoittanut: Ahti Syreeni

```

```

pvm:          23.4.2001

```

```

=====*/

```

```

public class Nuottilomake extends HttpServlet{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        /**generoi nuotinnlisäyslomakkeen*/

```

```

        ServletOutputStream palaute;

```

```

res.setContentType("text/html");
palaute= res.getOutputStream();
boolean rekisteroitunut=false;
Cookie[] piparit=req.getCookies();

//Rekisteroitunut käyttäjä?
for(int i=0;i<piparit.length;i++){
    if(piparit[i].getName().equals("nuottiarkisto")){
        if(!(piparit[i].getValue()==null))
            rekisteroitunut=true;
    }
}

if(!rekisteroitunut){ //toiminto vaatii rekisteröitymisen
    res.sendRedirect("Etusivu");
}
else{
    palaute.println(Nuottilomaketekstit.aluste());
    palaute.println(Nuottilomaketekstit.linkit());

    //luodaan yhteys tietokantaan
    Connection yhteys=null;
    yhteys= Tietokanta.luoYhteys();
    if (yhteys==null) {
        palaute.println(Virheilmoitukset.tkYhteysvirhe());
        return;
    }

    //haetaan aihepiirien nimet tietokannasta
    Statement stmt = null;
    ResultSet hakutulos = null;
    String[] tulostaulukko;

    try {
        stmt = yhteys.createStatement();
        hakutulos = stmt.executeQuery("SELECT DISTINCT nimi FROM "+
            " aihepiiri ORDER BY nimi");

        tulostaulukko=new String[hakutulos.getFetchSize()];

        for(int i=0; i<hakutulos.getFetchSize();i++)
            if(hakutulos.next())
                tulostaulukko[i]=hakutulos.getString(1);

        palaute.println(Nuottilomaketekstit.lomake(tulostaulukko));
    }
}

```

```

        palaute.println(Html.lopetus());
    }catch (SQLException e){
        palaute.println(Virheilmoitukset.SQLVirhe(e.getMessage()));
        //ei pitäisi koskaan tapahtua...
    }finally{
        try {
            if (hakutulos!=null) hakutulos.close();
            if (stmt!=null) stmt.close();
            Tietokanta.suljeYhteys(yhteys);
        } catch(SQLException e) {}
    }
}
}
}
}

```

```

/*****
Kirjastoluokka Nuottiarkiston nuottilisäyslomakkeen tekstien tallentamiseen
Kaikki tekstit ovat html-muodossa ja luokan metodit palauttavat
tekstit html-muodossa.

```

kirjoittanut: Ahti Syreeni

pvm: 26.4.2001

=====\*/

```

public class Nuottilomaketekstit{
    /*******
    private static final String tausta=
        "http://db.cs.helsinki.fi/u/asyreeni/tausta.jpg";
    /*******
    public static String aluste(){
        /**palauttaa nuottilisäyslomakesivun alusteen. Aluste sisältää myös
        lyhyet ohjeet*/

        return Html.aluste("Nuottiarkisto -nuottijulkaisun lis&auml;ys",tausta)+
            Html.otsikko("Nuottijulkaisun lis&auml;ys arkistoon",1)+"\n"+
            "<p>Voit lis&auml;t&auml; l&ouml;yt&auml;mi&auml;si"+
            " nuottijulkaisuja nuottiarkistoon muiden\
            n iloksi. Muista"+
            " kuitenkin olevasi vastuussa siit&auml;, ett&auml; teos on"+
            " luvallisesti esill&auml;. Nuottiarkistoon ei saa lis&auml;"+
            " t&auml; linkkej&auml; tekij&auml;noikeuksia rikkoviin"+
            " sivustoihin.<strong>Syötteissä ei saa olla hipsuja."+
            " </strong>Ylempi s&auml;velt&auml;j&auml;kentt&auml; on"+
            " s&auml;velt&auml;j&auml;n etunimelle,"+
            " alempi sukunimelle. Aihepiirilistä on monivalittava ja voit\n"+

```

```

    " my&ouml;s lis&auml;t&auml; oman aihepiirin. \n"+
    "</p>";
}
//-----
public static String lomake(String[] aihepiirit){
    /**palauttaa koko lomakkeen annetuilla aihepiireillä*/
    String palaute=lomakeA();

    if(aihepiirit==null)
        return "";

    palaute+=
        " <tr>\n"+
        " <td rowspan=\"2\"><label for=\"vaihepiiri\">Aihepiiri:+
            </label></td>\n"+
        " <td><select multiple=\"multiple\" name=\"vaihepiiri\" "+
            "size=\"4\" style=\"width:100%;\">"+
        " <option selected>(ei valittu)\n";

    if(aihepiirit.length>0){ //sitten annetut aihepiirit valikkoon
        for(int i=0;i<aihepiirit.length;i++)
            if(!(aihepiirit[i]==null))
                palaute+=" <option>"+aihepiirit[i]+" \n";
    }
    palaute+=" </select>\n";
    palaute+=lomakeB();
    return palaute;
}
//-----
public static String linkit(){
    /**palauttaa sivun linkit*/
    String palaute=
        "<p>"+Html.linkki("Etusivu ", "Etusivu")+ "--"+
        Html.linkki("Nuotinpoisto", "Nuotinpoistolomake")+ "--\n " +
        Html.linkki("Henkil&ouml;tietojen muuttaminen",
            "Rekisterointilomake")+ "--"+
        Html.linkki("Nuotinlis&auml;ys", "Nuottilomake")+ "--"+
        Html.linkki("Lopetus",
            "Logout")+ "\n"+

        "</p>";
    return palaute;
}
//-----
private static String lomakeA(){

```

```

/**palauttaa lomakkeen ennen vaihtuvia aihepiirejä olevat osat*/
return
" <form method=\"post\" action=\"Nuotinlisays\">\n"+
" <table border=1 style=\"width:100%;\">\n"+
" <col style=\"width:30%;\"><col style=\"width:70%;\">\n"+
" <tr>\n"+
" <td><label for=\"nimi\"><em>Teoksen nimi:</em></label>"+
" </td>\n"+
" <td><input type=\"text\" name=\"nimi\" "+
" style=\"width:100%;\"></td>\n"+
" </tr>\n"+
" <tr>\n"+
" <td><label for=\"saveltaja\">S&auml;velt&auml;j&auml;:"+
" </label></td>\n"+
" <td><label for=\"etunimi\">Etunimi:</label>\n"+
" <input type=\"text\" name=\"etunimi\" "+
" style=\"width:100%;\">\n"+
" <label for=\"sukunimi\">Sukunimi:</label>\n"+
" <input type=\"text\" name=\"sukunimi\" "+
" style=\"width:100%;\"></td>\n"+
" </tr>\n"+
" <tr>\n"+
" <td><label for=\"savellysvuosi\">S&auml;vellysvuosi:"+
" </label></td>\n"+
" <td><input type=\"text\" name=\"savellysvuosi\" "+
" style=\"width:100%;\" maxlength=\"4\"></td>\n"+
" </tr>\n";
}
//-----
private static String lomakeB(){
/**palauttaa lomakkeen aihepiirin jälkeiset osat*/
return
" <tr>\n"+
" <td><label for=\"aihepiiri\">Uusi aihepiiri:</label>\n"+
" <input type=\"text\" name=\"aihepiiri\" "+
" style=\"width:100%;\"></td>\n"+
" </tr>\n"+
" <tr>\n"+
" <td><label for=\"koko\">Koko:</label></td>\n"+
" <td><input type=\"text\" name=\"koko\" "+
" style=\"width:100%;\"></td>\n"+
" </tr>\n"+
" <tr>\n"+
" <td><em><label for=\"url\">URL-osoite:</label></em></td>\n"+
" <td><input type=\"text\" name=\"url\" "+

```

```

        "style=\"width:100%;\"></td>\n"+
" </tr>\n"+
" <tr>\n"+
"   <td><label for=\"lisatiedot\">Lisä tiedot:</label>"+
"     </td>\n"+
"   <td><textarea name=\"lisatiedot\" style=\"width:100%;\">"+
"     </textarea></td>\n"+
" </tr>\n"+
" <tr>\n"+
"   <td><input type=\"submit\" value=\"Lisä;het\">\n"+
"     <input type=\"reset\" value=\"Tyhjenn\"></td>\n"+
" </tr>\n"+
"</table>";
}
//-----
public static String lisattyKantaan(){
    /**palauttaa ilmoituksen siitä, että teos on onnistuneesti lisätty
    tietokantaan*/

    return Html.aluste("Nuottiarkisto -Nuottijulkaisu lisä;tty",
        tausta)+
        Html.otsikko("Nuottijulkaisu lisä;tty kantaan",1)+
        "\n"+linkit();
}

}

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

/*=====
Servletti Nuottijulkaisun lisäämiseen arkistoon

kirjoittanut: Ahti Syreeni
pvm:         23.4.2001
=====*/
public class Nuotinlisays extends HttpServlet{
    /*******
    //tunniste, jonka perusteella haetaan oikea pipari
    private static final String piparinimi="nuottiarkisto";
    private static final String

```

```

        tausta="http://.db.cs.helsinki.fi/u/asyreeni";
//*****

public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
    /**lisää nuottijulkaisun arkistoon*/

    ServletOutputStream palaute;
    res.setContentType("text/html");
    palaute= res.getOutputStream();
    String ktunnus=null;
    Cookie[] piparit=req.getCookies();

    //onko pakolliset tiedot, teoksen nimi ja url, annettu...
    if(req.getParameter("nimi").equals("") ||
        req.getParameter("nimi")==null ||
        req.getParameter("url").equals("")){
        palaute.println(Virheilmoitukset.eiNuottitietoja());
        return;
    }

    //tunnistetaan käyttäjä piparista
    for(int i=0;i<piparit.length;i++){
        if(piparit[i].getName().equals("nuottiarkisto")){
            if(!piparit[i].getValue()==null){
                ktunnus=piparit[i].getValue();
            }
        }
    }

    //onko teoksen saveltaja jo tietokannassa? Laaditaan SQL-kysely!
    String kysely="";

    if(!(req.getParameter("sukunimi").equals(""))){
        kysely=" sSukunimi='"+req.getParameter("sukunimi")+"'";
    }
    if(!(req.getParameter("etunimi").equals(""))){
        if(!(req.getParameter("sukunimi").equals(""))){
            kysely+=" AND ";
        }
        kysely+=" sEtunimi='"+req.getParameter("etunimi")+"'";
    }
    kysely="SELECT saveltajanumero FROM saveltaja WHERE "+kysely;

    Connection yhteys=null;
    yhteys= Tietokanta.luoYhteys();
    Statement stmt = null;

```

```

try {

stmt = yhteys.createStatement();
ResultSet hakutulos = null;

if (yhteys==null) {
    palaute.println(Virheilmoitukset.tkYhteysvirhe());
    if (stmt!=null) stmt.close();
    Tietokanta.suljeYhteys(yhteys);
    return;
}

int saveltajanumero=-1;

if(!((req.getParameter("sukunimi").equals("") &&
    req.getParameter("etunimi").equals(""))) ){
    hakutulos=stmt.executeQuery(kysely);
    if(hakutulos.next())
        saveltajanumero=hakutulos.getInt("saveltajanumero");
}

//jos uusi säveltäjä, lisätään tietokantaan
if(saveltajanumero!=-1){
    kysely="INSERT INTO saveltaja VALUES (saveltajatunniste."+
        "NEXTVAL, ";
    kysely+="'" +req.getParameter("etunimi)+"', ";
    kysely+="'" +req.getParameter("sukunimi)+"' )";
    stmt.executeUpdate(kysely);
}

//onko teos jo tietokannassa?
kysely="SELECT tunnistenumero FROM teos WHERE "+
    "saveltaja="+saveltajanumero;
if(!(req.getParameter("nimi").equals("")))
    kysely+=" AND tNimi='"
        +req.getParameter("nimi)+"'";
if(!(req.getParameter("savellysvuosi").equals("")))
    kysely+=" AND savellysvuosi="
        +req.getParameter("savellysvuosi");
if(!(req.getParameter("koko").equals("")))
    kysely+=" AND koko="
        +req.getParameter("koko");
if(!(req.getParameter("url").equals("")))
    kysely+=" AND URLosoite='"
        +req.getParameter("url)+"'";
if(!(req.getParameter("lisatiedot").equals("")))
    kysely+=" AND lisatiedot='"

```



```

        +hipsutPois(req.getParameter(
                    "lisatiedot"))+"';
//jos teos jo kannassa, voidaan katsoa se lisätyksi
hakutulos=stmt.executeQuery(kysely);
if(hakutulos.next()){
    if (stmt!=null) stmt.close();
    Tietokanta.suljeYhteys(yhteys);
    palaute.println(Nuottilomaketekstit.lisattyKantaan());
    return;
}
//muuten lisätään teos kantaan
kysely="INSERT INTO teos VALUES (teostunniste.NEXTVAL, '"+
        req.getParameter("nimi")+"', '"+ktunnus+"', ";

//jos säveltäjä on juuri lisätty kantaan, tietokannan
//saveltajanumero on sekvenssin tämän hetkinen arvo (ja metodin
//saveltajanumero=-1), muuten säveltäjännumero on vielä
//muuttujassa saveltajanumero

if(saveltajanumero==-1)
    kysely+="saveltajatunniste.CURRVAL, ";
else
    kysely+="saveltajanumero+", ";

if(!(req.getParameter("savellysvuosi").equals("")))
    kysely+=req.getParameter("savellysvuosi")+", ";
else
    kysely+="null, ";

if(!(req.getParameter("koko").equals("")))
    kysely+=req.getParameter("koko")+", ' ";

else
    kysely+="null, '";

kysely+=req.getParameter("url")+
        "', '+hipsutPois(req.getParameter("lisatiedot"))+
        "', sysdate)";
stmt.executeUpdate(kysely);

int aihenumero=-1;

//onko lisätty uusi aihepiiri, jota ei ole tietokannassa?
if(!(req.getParameter("aihepiiri").equals(""))){
    //onko lisätty aihepiiri jo tietokannassa?

```

```

kysely="SELECT aihenumero FROM aihepiiri "+
        "WHERE nimi='"+req.getParameter("aihepiiri")+"'";
hakutulos=stmt.executeQuery(kysely);

if(hakutulos.next()){ //oli jo tietokannassa, ei lisätä
    aihenumero=hakutulos.getInt("aihenumero");
}
else{ //lisätään uusi aihepiiri
    kysely="INSERT INTO aihepiiri VALUES ("+
            "aihepiiritunniste.NEXTVAL, '"+
            req.getParameter("aihepiiri")+"'";
    stmt.executeUpdate(kysely);
}
//liitetään aihepiiri teokseen yhteystaulun avulla
kysely="INSERT INTO aiheyhteys VALUES (teostunniste.CURRVAL,";
if(aihenumero!=-1)
    kysely+="aihepiiritunniste.CURRVAL)";
else
    kysely+="aihenumero)";

    stmt.executeUpdate(kysely);
}
//oliko valittu aihepiirejä vaihtoehtoista??
if(!(req.getParameterValues("vaihepiiri")==null)){
    for(int i=0;i<req.getParameterValues("vaihepiiri").length;i++){
        if(!(req.getParameterValues("vaihepiiri")[i].equals("ei"+
            "valittu"))){
            kysely="INSERT INTO aiheyhteys "+
                    "(SELECT teostunniste.CURRVAL, aihenumero FROM "+
                    "aihepiiri WHERE nimi='"+req.getParameterValues(
                    "vaihepiiri")[i]+"'";
            stmt.executeUpdate(kysely);
        }
    }
}
palaute.println(Nuottilomaketekstit.lisattyKantaan());
}catch (SQLException e){
    palaute.println(Virheilmoitukset.kielletytMerkit());
}finally{
    try {
        if (stmt!=null) stmt.close();
        Tietokanta.suljeYhteys(yhteys);
    } catch(SQLException e) {}
}
}

```

```

}
//-----
private static String hipsutPois(String muutettava){
    /**poistaa hipsut annetusta merkkijonosta*/
    char[] apu;
String palautettava="";
try{
    if(muutettava==null || muutettava.equals(""))
return muutettava;
apu=muutettava.toCharArray();
for(int i=0; i<apu.length;i++)
if(apu[i]!='\')
    apu[i]=' ';
return new String(apu);
}catch(Exception e){
return muutettava;
}
}
}

```

```

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

```

```

/*=====
Servletti nuottijulkaisun poistamisen valintalomakkeen generoimiseen

kirjoittanut: Ahti Syreeni
pvm:          23.4.2001
=====*/

```

```

public class Nuotinpoistolomake extends HttpServlet{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        ServletOutputStream palaute;
        res.setContentType("text/html");
        palaute= res.getOutputStream();
        String ktunnus=null;
        Cookie[] piparit=req.getCookies();

        //Rekisteroitynyt kayttaja?
        for(int i=0;i<piparit.length;i++){
            if(piparit[i].getName().equals("nuottiarkisto")){

```

```

        if(!(piparit[i].getValue()==null))
            ktunnus=piparit[i].getValue();
    }
}

if(ktunnus==null){
    res.sendRedirect("Etusivu");
}
else{
    palaute.println(Nuotinpoistotekstit.aluste());
    //luodaan yhteys tietokantaan
    Connection yhteys=null;
    yhteys= Tietokanta.luoYhteys();
    if (yhteys==null) {
        palaute.println(Virheilmoitukset.tkYhteysvirhe());
        return;
    }

    //katsotaan tietokannassa, onko käyttäjä arkistonhoitaja
    Statement stmt = null;
    ResultSet hakutulos = null;
    String kysely="";
    boolean arkistonhoitaja=false;

    try {
        stmt = yhteys.createStatement();
        kysely="SELECT DISTINCT oikeudet FROM kayttajatiedot"+
            " WHERE kayttajatunnus='"+ktunnus+"'";
        hakutulos = stmt.executeQuery(kysely);
        if(!hakutulos.next()){
            if (hakutulos!=null) hakutulos.close();
            if (stmt!=null) stmt.close();
            Tietokanta.suljeYhteys(yhteys);
            return;
        }
        if(!(hakutulos.getString("oikeudet")==null))
            arkistonhoitaja=true;

        if(!arkistonhoitaja)
            kysely="SELECT distinct tunnistenumero, tNimi, lisaaja, sSukunimi, "+
                " sEtunimi, URlosoite, lisayspaiva FROM teos, saveltaja "+
                " WHERE lisaaja='"+ktunnus+"' AND saveltaja=saveltajanumero "+
                " ORDER BY sSukunimi, sEtunimi,tNimi";

    }
    else

```

```

kysely="SELECT distinct kayttajatunnus, tunnistenumero, tNimi, "+
      " lisaaja, sSukunimi,"+
      " sEtunimi, URlosoite, lisayspaiva FROM teos, saveltaja, "+
      " kayttajatiedot WHERE lisaaja=kayttajatunnus AND "+
      " saveltaja=saveltajanumero GROUP BY kayttajatunnus,"+
      " tunnistenumero, tNimi, lisaaja, sSukunimi,"+
      " sEtunimi, URlosoite, lisayspaiva"+
      " ORDER BY kayttajatunnus, sSukunimi, sEtunimi, tNimi";

hakutulos = stmt.executeQuery(kysely);
palaute.println(Nuotinpoistotekstit.lomake());
palaute.println(Nuotinpoistotekstit.napit());

if(!arkistonhoitaja)
    palaute.println(Nuotinpoistotekstit.lomakeotsikko(false));
else
    palaute.println(Nuotinpoistotekstit.lomakeotsikko(true));

String lisaaja=null;

while(hakutulos.next()){
    if(arkistonhoitaja)
        lisaaja=hakutulos.getString("lisaaja");
    palaute.println(Nuotinpoistotekstit.teosrivi(
        hakutulos.getString("sSukunimi"),
        hakutulos.getString("sEtunimi"),
        hakutulos.getString("tNimi"),
        hakutulos.getString("lisayspaiva"),
        lisaaja,
        hakutulos.getInt("tunnistenumero"),
        hakutulos.getString("URlosoite")
    ));
}

palaute.println(Nuotinpoistotekstit.napit());
palaute.println(Nuotinpoistotekstit.lomakelopuke());
palaute.println(Nuotinpoistotekstit.linkit());

if(arkistonhoitaja){
    kysely="SELECT distinct kayttajatunnus, kEtunimi, kSukunimi,"+
          "sahkopostiosoite FROM kayttajatiedot ORDER BY "+
          "kSukunimi, kEtunimi";
    hakutulos = stmt.executeQuery(kysely);

```



```

        "<p>Valitse poistettavat nuottijulkaisut "+
        "   merkitsem&auml;ll&auml; s&auml;velt&auml;j&auml;n "+
        "   eteen rasti. Lopuksi l&auml;het&auml; lomake.";
    }
//-----
public static String napit(){
    /**palauttaa nuotinpoistolomakkeen napit*/
    return
        "<td><input type=\"submit\" value=\"Poista valitut arkistosta\">"+
        "</td>\n"+
        "<td><input type=\"reset\" value=\"Tyhjenn&auml;\"></td>\n";
    }
//-----
public static String lomake(){
    /**palauttaa lomakkeen alusteen*/
    return
        "<form method=\"post\" action=./Nuotinpoisto>\n"+
        "Html.taulukkoaluste(0,\"style=\"width:100%;\"");
    }
//-----
public static String lomakeotsikko(boolean ktunnuksella){
    /**palauttaa nuotinpoistolomakkeen sarakkeiden nimet*/
    String palaute;
    palaute=" <tr>\n"+
        "   <th>S&auml;velt&auml;j&auml;</th>\n"+
        "   <th>Nuottijulkaisu</th>\n"+
        "   <th>Url-osoite</th>"+
        "   <th>Lis&auml;ysp&auml;iv&auml;</th>\n";
    if(ktunnuksella)
        palaute+=" <th>Lis&auml;&auml;j&auml;</th>\n";
    palaute+=" </tr>\n";

    return palaute;
}
//-----
public static String teosrivi(String sukunimi, String etunimi,
                             String teosnimi, String lisaysaika,
                             String lisaaja, int tunniste,
                             String url){
    /**palauttaa yhden rivin nuotinpoistolomakkeesta*/
    //ei null-arvoja tulosteeseen.
    String palaute;
    if(etunimi==null)
        etunimi="";
    else

```

```

        if(etunimi.equals("null"))
            etunimi="-";
    if(sukunimi==null)
        sukunimi="";
    else
        if(sukunimi.equals("null"))
            sukunimi="-";

    palaute=" <tr>\n"+
        " <td><input type=\"checkbox\" name="+tunniste+">"+
        " <label for=\""+sukunimi+", "+etunimi+">"+
        sukunimi+", "+etunimi+"</label></td>\n"+
        " <td>"+teosnimi+"</td>"+
        " <td>"+Html.linkki(url,url)+"</td>"+
        " <td>"+lisaysaika+"</td>\n";

    if(!(lisaaaja==null)) //kysymyksessä arkistonhoitajan lomake
        palaute+=" <td>"+Html.linkki(lisaaaja,"#+lisaaaja)+"</td>\n";
    palaute+=" </tr>\n";
    return palaute;
}

//-----
public static String kayttajatietotaulukko(){
    /**palauttaa alusteen käyttäjätietotaulukolle*/
    return Html.otsikko("K&auml;ytt&auml;j&auml;tiedot",2)+
        Html.taulukkoaluste(0,"style=\"width:100%;\"")+
        " <tr>\n"+
        " <th>Sukunimi</th>\n"+
        " <th>Etunimi</th>\n"+
        " <th>S&auml;hk&ouml;postiosoite</th>\n"+
        " <th>K&auml;ytt&auml;j&auml;tunnus</th>\n"+
        " </tr>\n";
}

//-----
public static String kayttajatietorivi(String ktunnus, String sukunimi,
        String etunimi, String sposti){
    /**palauttaa yhden rivin käyttäjätietotaulukkoon annetuilla
        parametreilla*/

    String palaute;
    if(etunimi==null)
        etunimi="";
    if(sposti==null)
        sposti="";

```



```

return
    "    <tr>\n"+
    "    <td>"+sukunimi+"</td>\n"+
    "    <td>"+etunimi+"</td>\n"+
    "    <td>"+Html.email(sposti, sposti)+"</td>\n"+
    "    <td>"+Html.nimea(ktunnus, ktunnus)+"</td>\n"+
    "    </tr>\n";
}
//-----
public static String lomakelopuke(){
    /**palauttaa lomakkeen lopputagin*/
    return " </table>\n </form>";
}
//-----
public static String linkit(){
    /**palautta linkit muille sivuille*/
    String palaute=
        "<p>"+Html.linkki("Etusivu ", "Etusivu")+ "--"+
        Html.linkki("Nuotinlisays", "Nuottilomake")+ "--"+
        Html.linkki("Nuotinpoisto", "Nuotinpoistolomake")+ "--\n  "+
        Html.linkki("Lopetus", "Logout")+ "\n"+
        "</p>";
    return palaute;
}
}

```

```

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
/*=====
Servletti nuottijulkaisun poistamiseen Nuottiarkistosta

kirjoittanut: Ahti Syreeni
pvm:          23.4.2001
=====*/

public class Nuotinpoisto extends HttpServlet{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        /**metodi poistaa valitun nuottijulkaisun tietokannasta teoksen
        tunnistenumeron perusteella*/

```

```

ServletOutputStream palaute;
res.setContentType("text/html");
palaute= res.getOutputStream();
String ktunnus=null;
Cookie[] piparit=req.getCookies();
String tuhottava;
String kysely="";
Statement stmt = null;
Enumeration poistettavat=req.getParameterNames();

//Rekisteroitynyt kayttaja?
for(int i=0;i<piparit.length;i++){
    if(piparit[i].getName().equals("nuottiarkisto")){
        if(!(piparit[i].getValue()==null))
            ktunnus=piparit[i].getValue();
    }
}

if(ktunnus==null){
    res.sendRedirect("Etusivu");
    return;
}

if(!poistettavat.hasMoreElements()){ //ei ollut valittu poistettavia
    res.sendRedirect("Nuotinpoistolomake");
    return;
}

//lisätään SQL-kyselyyn valittujen teosten tunnistenumerot
//poistolomakkeen check-boxit oli nimetty teosten tunnistenumeroilla

try{
    kysely="DELETE from teos where tunnistenumero="+
        (String)poistettavat.nextElement();
    while(poistettavat.hasMoreElements()){
        kysely+=" OR tunnistenumero="+ (String)poistettavat.nextElement();
    }
}catch(Exception e){
    res.sendRedirect("Nuotinpoistolomake");
    return;    //tätä ei koskaan pitäisi tapahtua
}

//luodaan yhteys tietokantaan ja suoritetaan poisto
Connection yhteys=null;
try{

```

```

yhteys= Tietokanta.luoYhteys();
if (yhteys==null) {
    palaute.println(Virheilmoitukset.tkYhteysvirhe());
    return;
}
stmt = yhteys.createStatement();
stmt.executeUpdate(kysely);
res.sendRedirect("Nuotinpoistolomake");
}catch(SQLException e){
    palaute.println(Virheilmoitukset.SQLVirhe(e.getMessage()));
}finally{
    try{
        if (stmt!=null) stmt.close();
        Tietokanta.suljeYhteys(yhteys);
    } catch(SQLException e) {}
}
}
}
}

```

```

import java.io.*;
import java.sql.*;
/*=====
Kirjastoluokka Nuottiarkiston tietokanta-asetusten tallentamiseen.
Tämän luokan asetuksia muuttamalla voidaan vaihtaa käytettävää
tietokantaa.

kirjoittanut: Ahti Syreeni
pvm:          23.4.2001
=====*/
public class Tietokanta{
    /*******tietokanta-asetukset*****
    private static final String tkAjuri="oracle.jdbc.driver.OracleDriver";
    private static final String
        tkServeri="jdbc:oracle:thin:@kontti.helsinki.fi:1522:tktb";
    private static final String tkKayttajatunnus="ops$asyreeni";
    private static final String tkSalasana="infolab";

    /*******
    public static String tkAjuri(){
        /**palauttaa käytettävän tietokanta-ajurin*/
        return tkAjuri;
    }
}
//-----

```

```

public static String tkServeri(){
    /**palauttaa käytettävän tietokantaserverin*/
    return tkServeri;
}
//-----
public static String tkKayttaja(){
    /**palauttaa käytettävän tietokannan käyttäjätunnuksen*/
    return tkKayttajatunnus;
}
//-----
public static String tkSalasana(){
    /**palauttaa käytettävän tietokantasalasanan*/
    return tkSalasana;
}
//-----
public static Connection luoYhteys(){
    /**palauttaa tietokantayhteyden ylhäällä määriteltyyn tietokantaan*/
    Connection yhteys=null;
    try{
        Class.forName(tkAjuri);
        yhteys = DriverManager.getConnection(tkServeri,
            tkKayttajatunnus,tkSalasana);
    }catch (Exception e) {}
    return yhteys;
}
//-----
public static void suljeYhteys(Connection yhteys) {
    /**sulkee tietokantayhteyden käytettyyn tietokantaan*/
    try {
        yhteys.close();
    }catch (Exception e) {}
}
}

```

```
import Html;
```

```

/*=====
Kirjastoluokka Nuottiarkiston virheilmoitusten tallentamiseen
Kaikki tekstit ovat html-muodossa ja luokan metodit palauttavat
tekstit html-muodossa.

```

```

kirjoittanut: Ahti Syreeni
pvm:          30.4.2001
=====*/
public class Virheilmoitukset{
    /*******
    //osoite, johon lähetetään odottamattomat virheilmoitukset
    private static final String toimittaja=
        " <p>\n"+
        "   Ahti Syreeni <br>\n"+
        "   050-3673368 <br>\n"+
        "   "+Html.email("ahti.syreeni@helsinki.fi",
                       "ahti.syreeni@helsinki.fi")+"\n"+
        " </p>\n";
    /*******
    //käytettävä taustakuva
    private static final String tausta=
        "http://db.cs.helsinki.fi/u/asyreeni/tausta.jpg";
    /*******
    public static String tkYhteysvirhe(){
        /**palauttaa tietokantayhteysvirheilmoituksen*/
        String virheselite=
            "Tietokanta yhteys on t&auml;ll&auml; hetkell&auml; poikki, joten"+
            "\n   t&auml;m&auml;n ohjelman k&auml;tt&auml; ei ole \n"+
            "   mahdollista. Ole hyv&auml; ja yrit&auml; my&auml;hemmin\n"+
            "   uudelleen.";

            return virheilmoitus("Tietokanta yhteys poikki",virheselite,null);
        }
    //-----
    public static String SQLVirhe(String virhe){
        /**palauttaa SQL-virheilmoituksen*/
        String virheselite=
            "Tapahtui odottamaton ohjelmoinnista johtuva virhe";

            return virheilmoitus("Virhe",virheselite,virhe);
        }
    //-----
    public static String eiNumero(){
        /**palauttaa ilmoituksen virheellisestä numeroarvosta*/
        String virheselite=
            "Antamasi lomakkeen jossakin kent&auml;ss&auml; on "+
            "virhe. Ole hyv&auml; ja tarkista antamasi "+
            "arvot: numerokent&auml;ss&auml; saa olla vain numeroita ja "+
            "hipsuja ei saa olla miss&auml;&auml;n kent&auml;ss&auml;";
        return virheilmoitus("Virheellinen arvo",virheselite,null);
    }
}

```

```

}
//-----
public static String salasanavirhe(){
    /**palauttaa virheilmoituksen väärästä salasanasta*/
    String virheselite=
        "Antamassasi salasanassa on virhe. Molempien antamiesi salasanojen "+
        "on oltava t&auml;sm&auml;llään samat. Ole hyv&auml; ja tarkista "+
        "antamasi salasanat.";
    return virheilmoitus("Virheellinen salasana",virheselite,null);
}
//-----
public static String ktunnus(){
    /**palauttaa virheilmoituksen käyttäjätunnuksen puuttumisesta*/
    String virheselite=
        "Et antanut k&auml;ytt&auml;j&auml;tunnusta. Se on annettava "+
        "Palaa takaisin ja anna kunnollinen k&auml;ytt&auml;j&auml;tunnus.";
    return virheilmoitus("K&auml;ytt&auml;j&auml;tunnus puuttuu",
        virheselite,null);
}
//-----
public static String ktunnusLaiton(){
    /**palauttaa virheilmoituksen käyttäjätunuksesta, jossa kiellettyjä
    merkkejä*/
    String virheselite=
        "Antamasi k&auml;ytt&auml;j&auml;tunnus sis&auml;lt&auml;&auml;"+
        "kiellettyj&auml; merkkej&auml;."+
        "K&auml;ytt&auml;j&auml;tunnus ei saa sis&auml;lt&auml;&auml; "+
        "erikoismerkkej&auml;, "+
        "kuten \"*!#%&\'`. Ole hyv&auml; ja anna uusi "+
        "k&auml;ytt&auml;j&auml;tunnus";
    return virheilmoitus("Virheellinen k&auml;ytt&auml;j&auml;tunnus",
        virheselite,null);
}
//-----
public static String sukunimi(){
    /**palauttaa virheilmoituksen sukunimen puuttumisesta*/
    String virheselite=
        "Et antanut sukunime&auml;si. Se on annettava "+
        "Palaa takaisin ja anna sukunimesi.";
    return virheilmoitus("Sukunimi puuttuu",
        virheselite,null);
}
//-----
public static String eiPipareita(){

```

```

/**palauttaa virheilmoituksen siit&auml;, ettei tunniste piparia voi
    asettaa*/
String virheselite=
    "Selaimesi ei tue ev&auml;steit&auml; (Cookies) tai olet kytkenyt ne "+
    "pois p&auml;&auml;lt&auml;. Kytke ev&auml;steet p&auml;&auml;lle "+
    "ja yrit&auml; uudelleen";
return virheilmoitus("Ev&auml;steet eiv&auml;t toimi",
                    virheselite,null);
}
//-----
public static String login(){
    /**palauttaa virheilmoituksen väärästä käyttäjätunnuksesta tai
        salasanasta*/
    String virheselite=
        "Antamasi salasana tai k&auml;ytt&auml;j&auml;tunnus on "+
        "v&auml;&auml;rin tai et ole "+
        Html.linkki("rekister&ouml;itynyt.", "Rekisterointilomake")+
        "Yrit&auml; uudestaan.";
    return virheilmoitus("Sis&auml;&auml;nkirjoittautuminen ep&auml;onnistui",
                        virheselite,null);
}
//-----
public static String eiNuottitietoja(){
    /**palauttaa virheilmoituksen puuttuvasta tiedosta*/
    String virheselite=
        "Tarkista antaneesi teokselle nimen ja URL-osoitteen"+
        "palaa takaisin ja lis&auml;&auml; puuttuvat tiedot"+
        "\n";
    return virheilmoitus("Lomakkeessa puuttuvia pakollisia tietoja",
                        virheselite,null);
}
//-----
public static String kielletytMerkit(){
    /**palauttaa virheilmoituksen vääristä arvoista*/
    return eiNumero();
}
//-----
private static String virheilmoitus(String virheotsikko, String
                                    virheselitys, String SQLvirhe){
    /**generoi virheilmoituksen*/

    String palaute;
    palaute=Html.aluste("Virheilmoitus",tausta)+
        Html.otsikko(virheotsikko,1)+
        " <p>\n "+virheselitys+"\n </p>\n";
}

```

```

if(SQLvirhe!=null)
    palaute+=" <p>\n"+
        "    Tietokanta antoi seuraavanlaisen virheilmoituksen:\n"+
        "    <br>"+SQLvirhe+"<br>\n"+"T&auml;llaista ei pit&auml;isi"+
        "    tapahtua, joten ota yhteys ohjelman toimittajaan ja "+
        "    kerro\n, millaisessa toiminnassa t&auml;llainen virhe\n"+
        "    syntyi. Ohjelman toimittaja: \n"+toimittaja+"\n"+
        "    </p>\n";

palaute+=" <p>\n"+
    "    Voit palata takaisin selaimen back-napista, \n"+
    "    tai voit palata "+Html.linkki("etusivulle.", "Etusivu")+
    "    </p>\n"+Html.lopetus();

return palaute;
}
}

```

```

/*=====
Kirjastoluokka html-rakenteiden tuottamiseen.

```

kirjoittanut: Ahti Syreeni

pvm: 23.4.2001

```

=====*/
public class Html{
    private static final String dokumenttityyppi=
        "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//FI">";
    /*******
    public static String aluste(String sivun_otsikko){
        /**palauttaa html-sivun alusteen ilman taustakuvaa */
        String palaute;
        palaute=dokumenttityyppi+"\n"+
            "<html>\n"+
            "<head>\n"+
            "    <title>"+sivun_otsikko+"</title>\n"+
            "</head>\n"+
            "<body>\n";
        return palaute;
    }
    //-----
    public static String aluste(String sivun_otsikko, String tausta){
        /**palauttaa html-sivun alusteen taustakuvalla */
        String palaute;
        palaute=dokumenttityyppi+"\n"+

```



```

    "<html>\n"+
    "<head>\n"+
    " <title>"+sivun_otsikko+"</title>\n"+
    "</head>\n"+
    "<body background=\""+tausta+"\">\n";
    return palaute;
}
//-----
public static String lopetus(){
    /**palauttaa html-sivun lopetuksen */
    return "</body>\n"<html>";
}
//-----
public static String otsikkokuva(String teksti, String kuva){
    /**palauttaa html-kuvan ja vaihtoehtotekstin */
    return "<center><img src=\""+kuva+"\" alt=\""+
        teksti+"\"></center>\n";
}
//-----
public static String otsikko(String otsikko, int koko){
    /**palauttaa annetun otsikon annetulla koolla */
    if(koko>9)
        koko=9;
    if(koko<1)
        koko=1;
    return "<h"+koko+">"+otsikko+"</h"+koko+">\n\n";
}
//-----
public static String linkki(String otsikko, String osoite){
    /**palauttaa html-linkin */
    return "<a href=\""+osoite+"\">"+otsikko+"</a>";
}
//-----
public static String email(String otsikko, String osoite){
    /**palauttaa email-osoitteen */
    return "<a href=\"mailto:"+osoite+"\">"+otsikko+"</a>";
}
//-----
public static String nimea(String nimettävä, String nimi){
    /**nimeää annetun merkkijonon*/
    return "<a name=\""+nimi+"\">"+nimettävä+"</a>\n";
}
//-----
public static String taulukkoaluste(int reunuskoko, String tyyli){
    /**palauttaa taulukkomäärittelyn */

```

```

String palaute;
if(reunuskoko<0)
reunuskoko=0;
if(reunuskoko>3)
reunuskoko=3;

if(reunuskoko>0)
return "<table border="+reunuskoko+" "+tyyli+">\n";
else
return "<table "+tyyli+">\n";
}

//-----
public static String taulukkolopetus(){
    /**palauttaa taulukon lopputagin */
return "</table>\n";
}

//-----
public static String taulukkorivi(boolean aloitus){
    /**palauttaa taulukon rivin */
if(aloitus)
return "<tr>\n";
else
return "</tr>\n";
}

//-----
public static String taulukkoalkio(String alkio){
    /**palauttaa taulukon alkion */
return "<td>"+alkio+"</td>\n";
}

//-----
public static String kappale(String kappale){
    /**palauttaa kappaleen */
return "<p>"+kappale+"</p>";
}
}

```

## C Tietokannan asennuslauseet

```
CREATE SEQUENCE aihepiiritunniste;
```

```
CREATE SEQUENCE teostunniste;
```

```
CREATE SEQUENCE saveltajatunniste;
```

```
exit;
```

```
INSERT INTO kayttajatiedot VALUES ('ahoitaja','Arkiston', 'Hoitaja',  
'tähän@sähköpostiosoite','salasana','k');
```

# Nuottiarkisto

Nuottiarkisto on www-pohjainen nuottijulkaisujen tietojen säilönuspaikka. Arkistoon kerätään Internetissä saatavilla olevien nuottijulkaisujen tietoja ja linkit niihin. Nuottiarkistosta voidaan hakea nuottijulkaisuja erilaisin hakuehdoin ja nuottijulkaisuja voidaan sekä lisätä että poistaa. *Lisääminen ja poistaminen edellyttävät rekisteröitymistä.* Jos olet rekisteröitynyt, voit kirjoittautua sisään hakulomakkeen alla olevalla pienellä lomakkeella.

<input type="checkbox"/> Hakusana:	<input type="text"/>
<input type="checkbox"/> Teoksen nimi:	<input type="text"/>
<input type="checkbox"/> Säveltäjä:	Etinimi: <input type="text"/>
	Sukunimi: <input type="text"/>
<input type="checkbox"/> Aihepiiri:	Klassinen/instrumentaali <input type="text"/>
<input type="checkbox"/> Sävellysvuosi:	<input type="text"/> tasan <input type="text"/>
<input type="checkbox"/> Vain uutuuudet:	1 kk <input type="text"/>
<input type="button" value="Lähetä"/> <input type="button" value="Tyhjennä"/>	

## D Kuvat html-sivuista

Käyttäjätunnus: <input type="text"/>	Salasana: <input type="text"/>
Ei käyttäjätunnusta? <a href="#">Rekisteröidy.</a>	<input type="button" value="Lähetä"/> <input type="button" value="Tyhjennä"/>

### Ohjeita arkistosta hakuun

Jokainen hakuvaihtoehto, joka halutaan hakuperusteeksi, on merkittävä lomakkeen vasemmassa reunassa olevalla rastilla. Useita hakuvaihtoehtoja voidaan merkitä kerralla hakuvaihtoehtoiksi, jolloin haettavien nuottijulkaisujen on täytettävä kaikki merkityt hakukriteerit. Mikäli hakuperuste on merkitty, mutta sille ei ole annettu mitään arvoja, hakuperustetta ei huomioida haussa. Täytettyäsi hakulomakkeen suorita haku painamalla lomakkeen alhaalla olevaa "Lähetä"-nappia. Voit myös tyhjätä lomakkeen "Tyhjennä"-napista.

- Hakusana
  - Jos merkitsit tämän hakuperusteen ja annat jonkin sanan, nuottiarkistosta haetaan kaikki ne nuottijulkaisut, joiden säveltäjän nimessä, teoksen nimessä, URL-osoitteessa, aihepiireissä tai lisätiedoissa esiintyy antamasi sana. Antamasi sanan kirjainkoolla ei ole merkitystä ja hakusanan ei tarvitse olla kokonainen sana ja hakusana voi sijaita nuottijulkaisun tiedoissa myös sanan keskellä. Niinpä esimerkiksi antamalla hakusanaksi kirjain a, saadaan hakutulokseksi kaikki ne teokset, joiden edellä luetuissa tiedoissa a-kirjain esiintyy. Hakutuloksia voi siis rajoittaa antamalla pitempiä hakusanoja. **Hakusana ei saa sisältää hipsuja (')**
- Teoksen nimi
  - Hakuehdolla haetaan kaikki ne teokset, joiden nimen alussa esiintyy annettu nimi. **Teoksen nimi ei saa sisältää hipsuja (')**, kirjainkoolla ei ole väliä
- Säveltäjä
  - Hakuehdolla haetaan kaikki ne teokset joiden säveltäjän nimessä on annettu sana. Säveltäjän nimet eivät saa sisältää hipsuja ('). **Toinen nimistä voidaan olla tyhjä Huom! Joissakin selaimissa ei näy tekstejä, jotka ilmoittavat, että ylempi nimikenttä on säveltäjän etunimi ja alempi kenttä säveltäjän sukunimi.**
- Aihepiiri
  - Hakuehdolla haetaan kaikki ne teokset, jotka kuuluvat valittuun aihepiiriin. Sama teos voi kuulua useampaankin aihepiiriin.
- Sävellysvuosi
  - Hakuehdolla haetaan kaikki ne teokset, jotka on sävelletty annettuna vuonna, sitä ennen, tai sen jälkeen. Vuosiluku annetaan aina täydellisenä (esim.1995).
- Vain uutuuudet
  - Hakuehdolla haetaan kaikki ne teokset, jotka on lisätty enintään valittu kuukausimäärä sitten.

Kuva 2: Nuottiarkiston etusivu

# Nuottiarkisto

**Koska olet rekisteröitynyt, voit valita vapaasti seuraavista vaihtoehdoista. Muista kuitenkin olevasi vastuussa lisäämistäsi teoksista. Nuottiarkistoon ei saa lisätä linkejä tekijänoikeuksia rikkoviin nuottijulkaisuihin.**

[Nuotinsisäys](#)-- [Nuotinpoisto](#)-- [Henkilötietojen muuttaminen](#)-- [Lopetus](#)

<input type="checkbox"/> Hakusana:	
<input type="checkbox"/> Teoksen nimi:	
<input type="checkbox"/> Säveltäjä:	Etinimi:
	Sukunimi:
<input type="checkbox"/> Aihepiiri:	Klassinen/instrumentaali
<input type="checkbox"/> Sävellysvuosi:	<input type="text"/> tasan
<input type="checkbox"/> Vain uutuuudet:	1 kk
<input type="button" value="Lähetä"/> <input type="button" value="Tyhjennä"/>	

Kuva 3: Nuottiarkiston rekisteröityneiden käyttäjien etusivu. Sivulla lisäksi hakuohjeet, kuten normaalilla etusivulla.

## Nuottijulkaisun lisäys arkistoon

Voit lisätä löytämäsi nuottijulkaisuja nuottiarkistoon muiden iloksi. Muista kuitenkin olevasi vastuussa siitä, että teos on luvallisesti esillä. Nuottiarkistoon ei saa lisätä linkejä tekijänoikeuksia rikkoviin sivustoihin. **Syötteissä ei saa olla hupsuja.** Ylempi säveltäjäkenttä on säveltäjän etunimelle, alempi sukunimelle. Aihepiiristä on monivalittava ja voit myös lisätä oman aihepiiriin.

[Etusivu](#)-- [Nuotinpoisto](#)-- [Henkilötietojen muuttaminen](#)-- [Nuotinsisäys](#)-- [Lopetus](#)

Teoksen nimi:	
Säveltäjä:	Etinimi:
	Sukunimi:
Sävellysvuosi:	
Aihepiiri:	(ei valittu) Klassinen/instrumentaali Klassinen/orkesteri Klassinen/sinfonia
	Uusi aihepiiri:
Koko:	
URL-osoite:	
Lisätiedot:	
<input type="button" value="Lähetä"/> <input type="button" value="Tyhjennä"/>	

Kuva 4: Nuottijulkaisun lisäyssivu



## Nuottijulkaisun poisto arkistosta

Valitse poistettavat nuottijulkaisut merkitsemällä säveltäjän eteen rasti. Lopuksi lähetä lomake.

Säveltäjä		Nuottijulkaisu	Url-osoite	Lisäyspäivä
<input checked="" type="checkbox"/>	Heiniö, Mikko	Opus improbum	<a href="http://www.spede.fi">http://www.spede.fi</a>	2001-05-02 20:31:04.0
<input type="checkbox"/>	Mozart, W.A	Klassinen teos huilulle ja orkesterille	<a href="http://www.yahoo.com">http://www.yahoo.com</a>	2001-05-02 20:36:25.0

[Etusivu](#) - [Nuotinsisäys](#) - [Nuotinpoisto](#) - [Lopetus](#)

Kuva 5: Nuottijulkaisun poistosivu (rekisteröitynyt käyttäjä)

## Nuottijulkaisun poisto arkistosta

Valitse poistettavat nuottijulkaisut merkitsemällä säveltäjän eteen rasti. Lopuksi lähetä lomake.

Säveltäjä	Nuottijulkaisu	Url-osoite	Lisäyspäivä	Lisääjä
<input type="checkbox"/> Bach, J.S	Konsertto huilulle ja continuoalle	<a href="http://www.mp3.com">http://www.mp3.com</a>	2001-04-27 04:57:44.0	<a href="#">asyreeni</a>
<input type="checkbox"/> Kohonen, Keijo	Syntetisaattorifantasia 1	<a href="http://www.keijo_music.com">http://www.keijo_music.com</a>	2001-04-27 05:07:47.0	<a href="#">asyreeni</a>
<input type="checkbox"/> Kohonen, Keijo	Syntetisaattorifantasia 2	<a href="http://www.keijo_music.com">http://www.keijo_music.com</a>	2001-04-27 05:08:05.0	<a href="#">asyreeni</a>
<input type="checkbox"/> Kohonen, Keijo	Syntetisaattorifantasia 3	<a href="http://www.keijo_music.com">http://www.keijo_music.com</a>	2001-04-27 05:08:12.0	<a href="#">asyreeni</a>
<input type="checkbox"/> Sibelius, Jean	Sinfonia 2	<a href="http://www.sibelius.fi/">http://www.sibelius.fi/</a>	2001-04-27 04:42:16.0	<a href="#">asyreeni</a>
<input type="checkbox"/> Sibelius, Jean	Sinfonia 5	<a href="http://www.sibelius.fi/s5.zip">http://www.sibelius.fi/s5.zip</a>	2001-04-27 04:44:58.0	<a href="#">asyreeni</a>
<input type="checkbox"/> Strauss, Richard	Oboekonsertto	<a href="http://www.sofs.com">http://www.sofs.com</a>	2001-04-27 04:53:03.0	<a href="#">asyreeni</a>
<input type="checkbox"/> Heiniö, Mikko	Opus improbum	<a href="http://www.spede.fi">http://www.spede.fi</a>	2001-05-02 20:31:04.0	<a href="#">esimerkki</a>
<input type="checkbox"/> Mozart, W.A	Klassinen teos huilulle ja orkesterille	<a href="http://www.yahoo.com">http://www.yahoo.com</a>	2001-05-02 20:36:25.0	<a href="#">esimerkki</a>
<input type="checkbox"/> Kullervo, Oja	Suomi poppia	<a href="http://www.suomi24.fi">http://www.suomi24.fi</a>	2001-05-03 15:25:38.0	<a href="#">hoitaja</a>
<input type="checkbox"/> Saariaho, Kaia	Sinfonia 4	<a href="http://www.saariaho.com">http://www.saariaho.com</a>	2001-05-05 10:23:18.0	<a href="#">hoitaja</a>

[Etusivu](#) - [Nuotinsisäys](#) - [Nuotinpoisto](#) - [Lopetus](#)

### Käyttäjätiedot

Sukunimi	Etunimi	Sähköpostiosoite	Käyttäjätunnus
Heiskanen	Matti	<a href="mailto:matti@hki.fi">matti@hki.fi</a>	mheiskanen
Hoitaja	Arkiston	<a href="mailto:hoitaja@arkisto.fi">hoitaja@arkisto.fi</a>	hoitaja
Kekkonen			ekononen
Kohonen	Kajaja	<a href="mailto:kajaja@kohonen.fi">kajaja@kohonen.fi</a>	kkohonen
Meikäläinen	Matti	<a href="mailto:matti@maja.com">matti@maja.com</a>	esimerkki
Saarnen	Kajaja	<a href="mailto:ksaarnen@helsinki.fi">ksaarnen@helsinki.fi</a>	ksaarnen
Suominen	Kai	<a href="mailto:kai@suomi24.fi">kai@suomi24.fi</a>	ksuominen
Syreeni	Ahti	<a href="mailto:ahti.syreeni@helsinki.fi">ahti.syreeni@helsinki.fi</a>	asyreeni
Syreeni	Sami	<a href="mailto:sami.syreeni@helsinki.fi">sami.syreeni@helsinki.fi</a>	ssyreeni
Öhlberg	Joonas	<a href="mailto:johlberg@yahoo.com">johlberg@yahoo.com</a>	johl

[Etusivu](#) - [Nuotinsisäys](#) - [Nuotinpoisto](#) - [Lopetus](#)

Kuva 6: Nuottijulkaisun poistosivu (arkiston hoitaja)

## Rekisteröitymislomake

Ole hyvä ja anna henkilötietosi alla oleviin kenttiin. Käyttäjätunnus ja sukunimi ovat pakollisia.

**Käyttäjätunnus on enimmillään kymmenen kirjaimen mittainen. Salasana annetaan molempiin salasarakenttiin ja niiden on oltava täsmälleen samat. Ota huomioon, että arkistonhoitaja näkee salasanasi. Mitkään sivun kentät eivät saa sisältää hipsuja.** Lähetettyäsi lomakkeen, pääset rekisteröityneiden etusivulle.

Etinimi:	<input type="text"/>
Sukunimi:	<input type="text"/>
Sähköposti:	<input type="text"/>
Käyttäjätunnus:	<input type="text"/>
Salasana:	<input type="password"/>
Salasana:	<input type="password"/>
<input type="button" value="Lähetä"/>	<input type="button" value="Tyhjennä"/>

[Etusivu](#)

Kuva 7: Rekisteröitymissivu

## Henkilötietojen muuttaminen

Voit muuttaa kaikkia muita tietojasi, paitsi käyttäjätunnustasi. Voit myös poistaa henkilötietosi valitsemalla henkilötietojen poiston. Samalla kuitenkin poistuvat myös lisäämäsi teokset Nuottiarkesta.

Etinimi:	<input type="text" value="Arkiston"/>
Sukunimi:	<input type="text" value="Hoitaja"/>
Sähköposti:	<input type="text" value="hoitaja@arkisto.fi"/>
Käyttäjätunnus:	<input type="text" value="hoitaja"/>
Salasana:	<input type="password" value=""/>
Salasana:	<input type="password" value=""/>
<input type="checkbox"/> Henkilötietojen poisto	<input type="button" value="Lähetä"/> <input type="button" value="Tyhjennä"/>

[Etusivu](#) -- [Nuotinlisäys](#) -- [Nuotinpoistolomake](#) -- [Lopetus](#)

Kuva 8: Henkilötietojen muutossivu

## Nuottihaun tulokset

[Etusivu](#) - [Nuotinsays](#) - [Nuotinpoisto](#) - [Lopetus](#)

Säveltäjä	Teoksen nimi	Sävellysvuosi	Koko	URL-osoite
Alho, Maarit	<a href="#">Saaristolaiselämää</a>	2000	12	<a href="http://www.alho.com">http://www.alho.com</a>
Kullervo, Oja	<a href="#">Suomi poppia</a>	1980	145	<a href="http://www.suomi24.fi">http://www.suomi24.fi</a>

[Etusivu](#) - [Nuotinsays](#) - [Nuotinpoisto](#) - [Lopetus](#)

### Alho, Maarit: Saaristolaiselämää

Abckiria on kirjakielen perustajan ensimmäinen teos, lukemisen opas, aapinen. Yhtään ensimmäisen painoksen täydellistä kappaletta ei ole säilynyt meidän päiviimme. Säilyneistä katkelmista päätellen tämä kirjallisuutemme, lukutaitomme, kirjoitetun kielen perusta on ollut varsin vaatimaton, laajuudeltaan 16-sivuinen vihkonen. Laaja ei ollut toinenkaan painos (ilm. 1551), siinä oli 24 sivua. Sen tittelin alla oleva alkutervehdys Michael Agricola Christiano Salutem voi olla osoitus joko kaikille kristityille tai Agricolan pojalle Kristianille, joka ensimmäisen painoksen ilmestyessä oli 9-vuotias. Kumpikin tulkintamahdollisuus on yhtä puhutteleva. Agricola halusi opettaa poikansa lukemaan. Agricola halusi myös opettaa suomalaiset lukemaan. Hän onnistui. Suomalaiset ovat maailman lukutaitoisempia kansoja. Helppoa ei lukutaidon omaksuminen aina ole ollut, niin kuin lukkarin koulu Seitsemässä veljeksessä osoittaa. Ja luettu on myös pyrittävä ymmärtämään. (Mikael Agricola on suomen kirjakielen ja suomalaisen kirjallisuuden perustaja sekä uskonpuhdistaja. Agricola opiskeli Wittenbergissä mm. Lutherin johdolla ja valmistui maisteriksi 1539. Hän toimi Turun piispana vuodesta 1554, kuolessaan asti. Uskonpuhdistuksen toteutuessa Agricola käänsi ja julkaisi kansan kielellä välttämättömät teokset. Useat Agricolan käyttämistä ja keksimistä sanoista ovat edelleen käytössä mm. esikuva, hallitus, hengellinen jne.)

Lisätty: 2001-05-05 10:35:24.0

### Kullervo, Oja: Suomi poppia

Suomen pop-musiikin helmiä. Nuotit Encore-muodossa.

Lisätty: 2001-05-03 15:25:38.0

[Etusivu](#) - [Nuotinsays](#) - [Nuotinpoisto](#) - [Lopetus](#)

Kuva 9: Hakutulossivu

## E Suunnitteludokumentti