

Suunnitteludokumentti

Koskelo

Helsinki 15.12.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Tom Bertell
Johan Brunberg
Lauri Liuhto
Eeva Nevalainen
Harri Tuomikoski

Asiakas

Teemu Kerola

Johtoryhmä

Juha Taina
Turjo Tuohiniemi

Kotisivu

<http://www.cs.Helsinki.FI/group/koskelo/>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	12.10.2004	Ensimmäinen kehitysversio
0.9	3.11.2004	Julkaisukandidaatti
1.0	16.11.2004	Katselmoinnin perusteella korjattu

Sisältö

1	Johdanto	1
1.1	Järjestelmän yleiskuvaus	1
1.2	Termit	2
2	Arkkitehtuurin kuvaus	3
2.1	Sisäiset osajärjestelmät	3
2.1.1	Composer	4
2.1.2	Displayer	4
2.1.3	Analyser	5
2.2	Ulkoiset osajärjestelmät	6
2.2.1	Titokone	6
2.2.2	eAssari	6
3	Luokkarakenne ja jsp-sivut	7
3.1	Displayer-osajärjestelmä	7
3.1.1	CommonDisplayer implements DisplayerInterface	10
3.1.2	DynamicTTK91Displayer extends CommonDisplayer	10
3.1.3	FillInTTK91Displayer extends CommonDisplayer	11
3.1.4	StaticTTK91Displayer extends CommonDisplayer	11
3.1.5	TTK91DisplayerUtils	12
3.1.6	DisplayerServlet2	12
3.2	Composer-osajärjestelmä	12
3.2.1	StaticTTK91Composer	12
3.2.2	DynamicTTK91Composer	14
3.2.3	FillInTTK91Composer	14

3.2.4	TTK91TaskParser	14
3.2.5	TTK91SyntaxChecker	15
3.2.6	TTK91TaskCriteria	15
3.2.7	TTK91TaskFeedback	16
3.2.8	TTK91TaskOptions	18
3.2.9	TaskDefinitionController	20
3.3	Analysier-osajärjestelmä	21
3.3.1	Answer2	21
3.3.2	CommonAnalysier implements AnalysierInterface	21
3.3.3	StaticTTK91Analysier extends CommonAnalysier	21
3.3.4	FillInTTK91Analysier extends CommonAnalysier	23
3.3.5	DynamicTTK91Analysier extends CommonAnalysier	23
3.3.6	TTK91FeedbackComposer	24
3.3.7	TTK91AnalysierUtils	24
3.3.8	TaskBase implements AttributeCache	24
3.3.9	Control (Titokoneen rajapintaluokka)	24
4	Käyttöliittymä	25
4.1	Composer – tehtävän luonti ja muokkaus	25
4.1.1	Kriteerien syntaksi	25
4.1.2	Symbolien merkitykset kriteerimäärittelyssä	28
4.1.3	Dynaamisen tehtävän määrittelyn syntaksi	28
4.2	Displayer – tehtävän ratkaisu ja palaute	29
4.3	Analysier – vastauksen analysointi	29
5	Muutokset ja lisäykset ulkoisiin osajärjestelmiin	29

	iii
5.1 Lisäykset eAssariin	29
5.2 Lisäykset Titokoneeseen	34
Lähteet	34

1 Johdanto

Koskelo on Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojekti (kurssi 581260–4). Projektin tavoitteena on tuottaa järjestelmä, joka liittää yhteen aiemmin ohjelmistotuotantoprojekteissa tuotetun Titokoneen [Kr04b] ja jatkokehitetyn eAssarin [Ar04]. Ohjelmiston tarkoituksena on tarjota helppokäyttöinen ympäristö Tietokoneen toiminta -kurssin (kurssi 581305–6) laskuharjoitusten luomiseen ja automaattiseen tarkastamiseen.

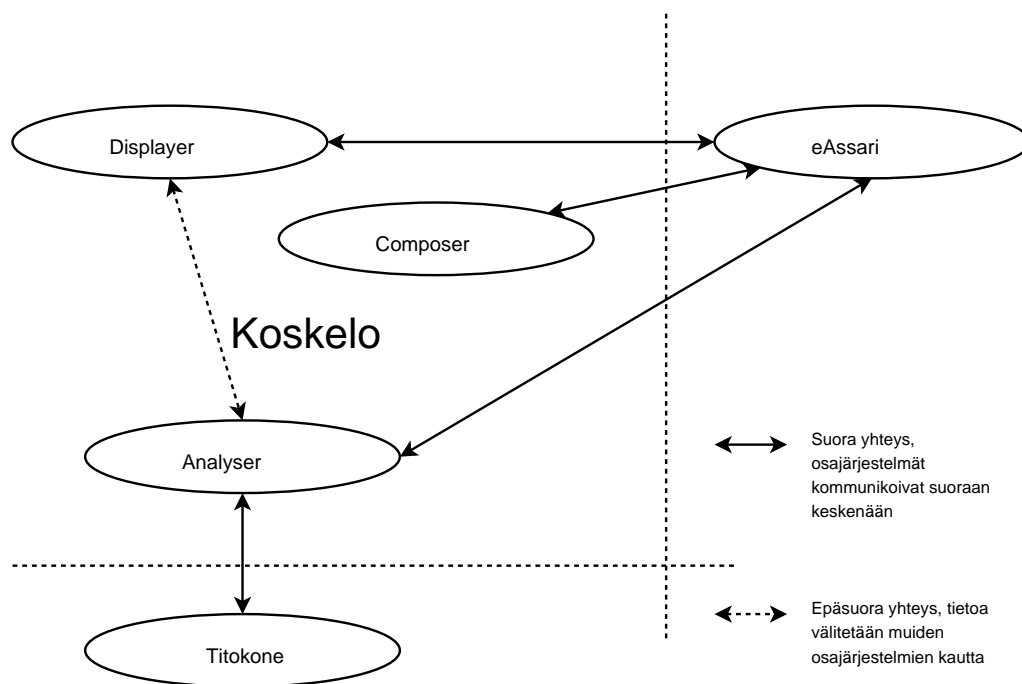
Tietokoneen toiminta -kurssilla käsitellään mm. kuvitteellisen TTK-91-koneen [Häk91] symbolista konekieltä, jonka simulaattori Titokone on. eAssari tarjoaa työkalut yksinkertaisten harjoitustehtävien automaattiseen tarkastamiseen. Koskelo-projektissa yhdistetään eAssariin Titokoneen toiminnallisuutta, jolloin TTK-91-kielisiä ohjelmia voidaan tarkastaa automaattisesti.

Projektin tuotokset julkaistaan GNU General Public Licensen tai GNU Lesser General Public Licensen [FSF99] alla Helsingin yliopiston valinnan mukaan. Lisenssitiedot ja projektin tuotokset, mukaanlukien tämä dokumentti, ovat saatavilla projektin kotisivulla, osoitteessa <http://www.cs.helsinki.fi/group/koskelo/>.

1.1 Järjestelmän yleiskuvaus

Järjestelmä on tarkoitettu tarjoamaan helppokäyttöinen liittymä, jolla voidaan luoda harjoitustehtäviä TTK-91-konekielen harjoitteluun eAssari-järjestelmässä, tarkastaa opiskelijoiden tehtäväsuorituksia ja käyttää Titokonetta tehtävien simulointiin sekä tarkastuksen apuna.

Kuva 1 esittää järjestelmän rakennetta karkealla tasolla. Ulkoinen osajärjestelmä eAssari toimii kehyksenä Koskelolle. Käyttäjä valitsee haluamansa toimenpiteet eAssarissa, joka käyttää toimintojen suorittamiseen Koskelon apua. Joissakin toiminnoissa (harjoitustehtävän tarkastaminen) Koskelo vuorostaan käyttää toisen ulkoisen osajärjestelmän Titokoneen palveluja.



Kuva 1: Järjestelmän yleinen rakenne

1.2 Termit

Järjestelmä Koskelo-projektin tuottamat ja käyttämät komponentit kokonaisuutena.

Kriteeri Ehto jolla tutkitaan opiskelijan ratkaisun oikeellisuutta ja laatua.

Tito Tietojenkäsittelytieteen laitoksen kurssi Tietokoneen toiminta (581305).

Titokone Ohjelmistotuotantoprojekti Kosken keväällä 2004 tuottama järjestelmä Koksi-simulaattorin korvaajaksi.

Koksi Titokoneen edeltäjä. Tietokoneen toiminta -kurssilla käytetty konekielen simulaattori.

TTK-91 Titokoneen käyttämä konekieli [Häk91].

Ohjelma Opiskelijan tai tehtävän laatijan antama Titokoneella suoritettava TTK-91-ohjelmakoodi.

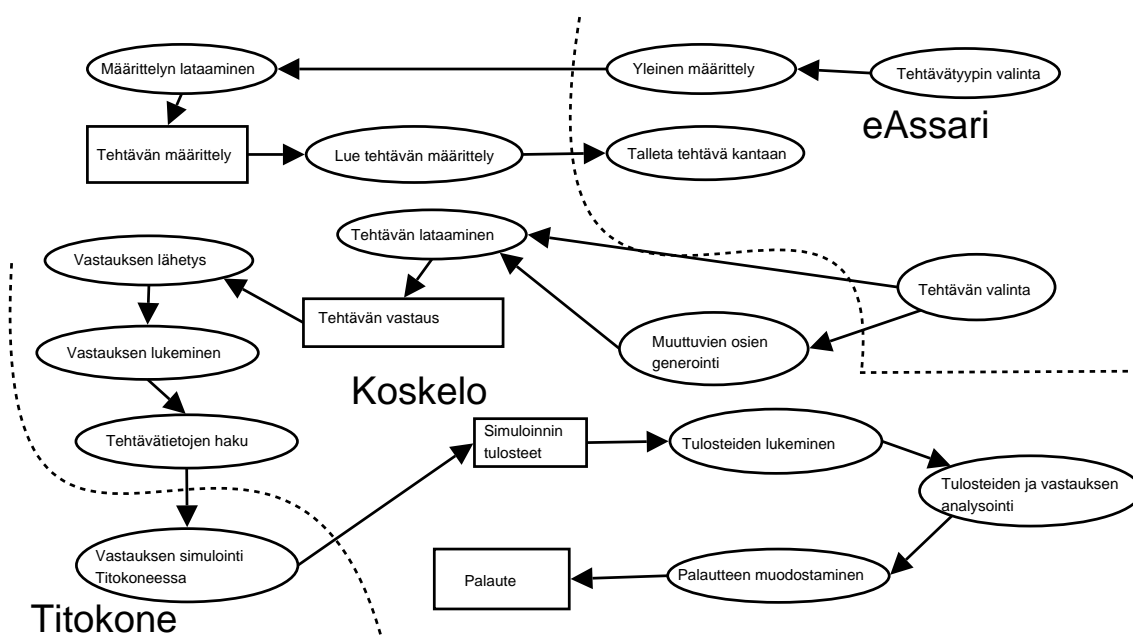
eAssari Automaattisesti tarkastettavat harjoitustehtävät mahdollistava järjestelmä, jota Koskelo-projekti laajentaa TTK-91-ohjelmointitehtäytyypeillä.

Assari eAssari-järjestelmää jatkokehittänyt ohjelmistotuotantoprojekti. Assari on siis osa eAssaria.

2 Arkkitehtuurin kuvaus

Järjestelmään kuuluu sisäisiä ja ulkoisia osajärjestelmiä. Ulkoisilla osajärjestelmillä tarkoitetaan valmiita Koskelo-ryhmästä riippumattomia ohjelmistokomponentteja, joita käytetään valmiiden rajapintojen avulla. Kuva 2 antaa käsityksen järjestelmän yleisestä rakenteesta sekä tiedon kulusta järjestelmän osien välillä.

Uuden tehtävän määrittely aloitetaan eAssarissa, joka käyttää Koskelon järjestämiä palveluja tehtävän luomisessa. Valmiin tehtävän ratkaisua yrittävä opiskelija aloittaa työn valitsemalla tehtävän eAssarin avulla, jonka jälkeen kontrolli siirtyy jälleen Koskelolle. Tehtävän ratkaisu tarkastetaan vertaamalla Titokoneella simuloidun ohjelman tulosta tarkastuskriteereihin. Suoritustiedot tallennetaan eAssariin. Tehtävän oikeellisuustarkastuksen jälkeen opiskelijan ratkaisusta annetaan palautetta.



Kuva 2: Tiedon siirtyminen järjestelmässä

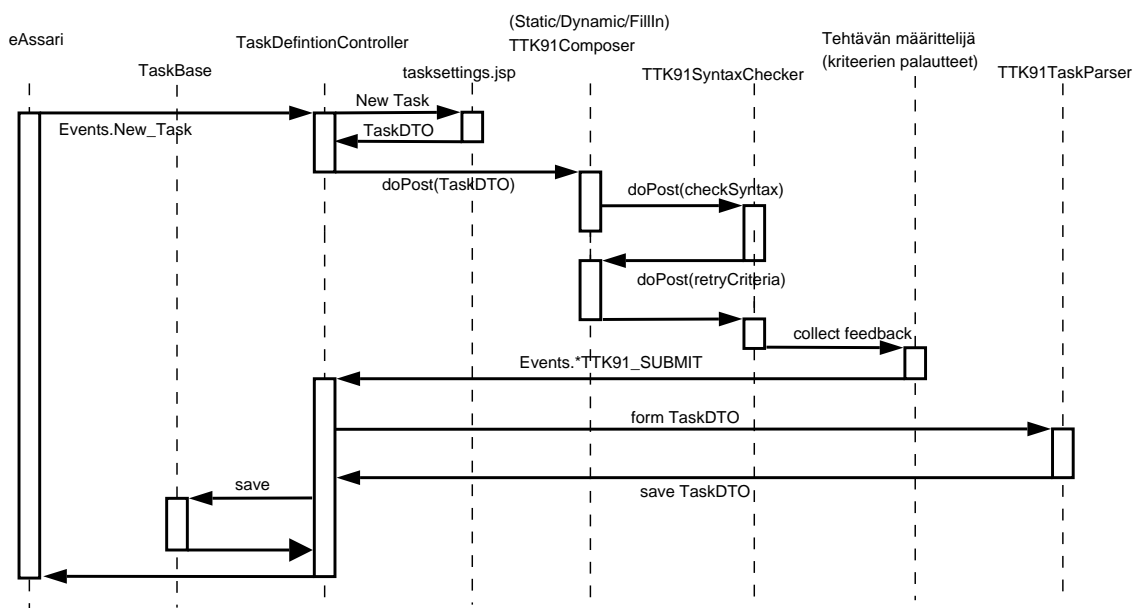
2.1 Sisäiset osajärjestelmät

Järjestelmä rakentuu kolmen sisäisen osajärjestelmän varaan. Ne ovat *Composer*, *Displayer* ja *Analyser*.

2.1.1 Composer

Composer-osajärjestelmä on tehtävien luontiin ja muokkaukseen liittyvien palvelujen tarjoaja. Ulkoinen osajärjestelmä eAssari kutsuu Composeria kun järjestelmällä luodaan uusi tehtävä tai muokataan vanhaa. Jokaista tehtävätyyppiä (staattinen, dynaaminen ja täydennys)varten on oma TTK91Composer.

Composer tarjoaa käyttäjälle lomakkeen, jolla voidaan syöttää tehtävänanto. Tämän lisäksi Composer tarkistaa tehtävänannon syntaksin oikeellisuuden ja palauttaa valmiin tehtävän eAssarille tallennettavaksi tietokantaan. Composerin tehtävänmäärittelyä esittävä sekvenssikaavio on kuvassa 3.

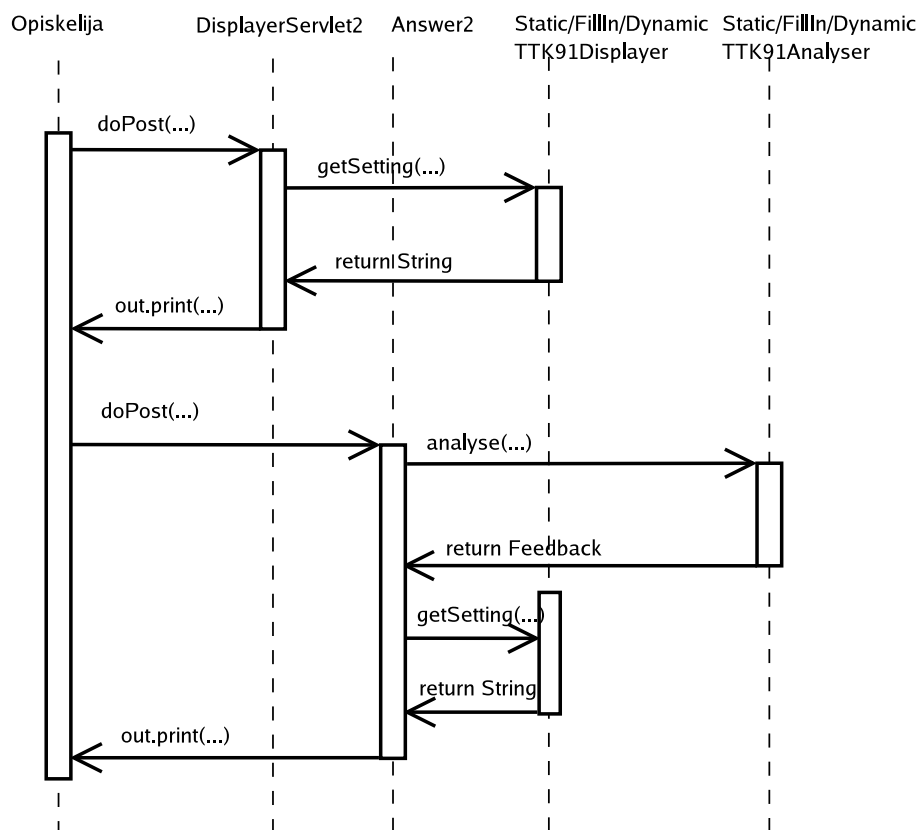


Kuva 3: Tehtävänmäärittelyn sekvenssikaavio

2.1.2 Displayer

Displayer-osajärjestelmä muodostaa opiskelijan rajapinnan järjestelmään. Se rakentaa html-sivun, jolla on tehtävänanto ja vastauslomake. Jokaisella tehtävätyypillä on oma TTK91Displayer-luokka. Ratkaisuyrityksen jälkeen eAssari liittyy sivulle myös edelliseen yritykseen liittyvän palautteen: onko vastaus oikein, tehtävänlaajan kommentit, Titokoneen tila ja tulosteet, käytettyjen konekäskyjen määrä sekä suorituksen statistiikka.

Displayer on vuorovaikutuksessa eAssarin kanssa. eAssari pyytää Displayerilta tehtävänannon, johon opiskelija syöttää vastauksensa. Analyser tarkastaa vastauksen, ja muodostaa siihen liittyvän palautteen. Kuvassa 4 esitellään Displayerin toimintaa.

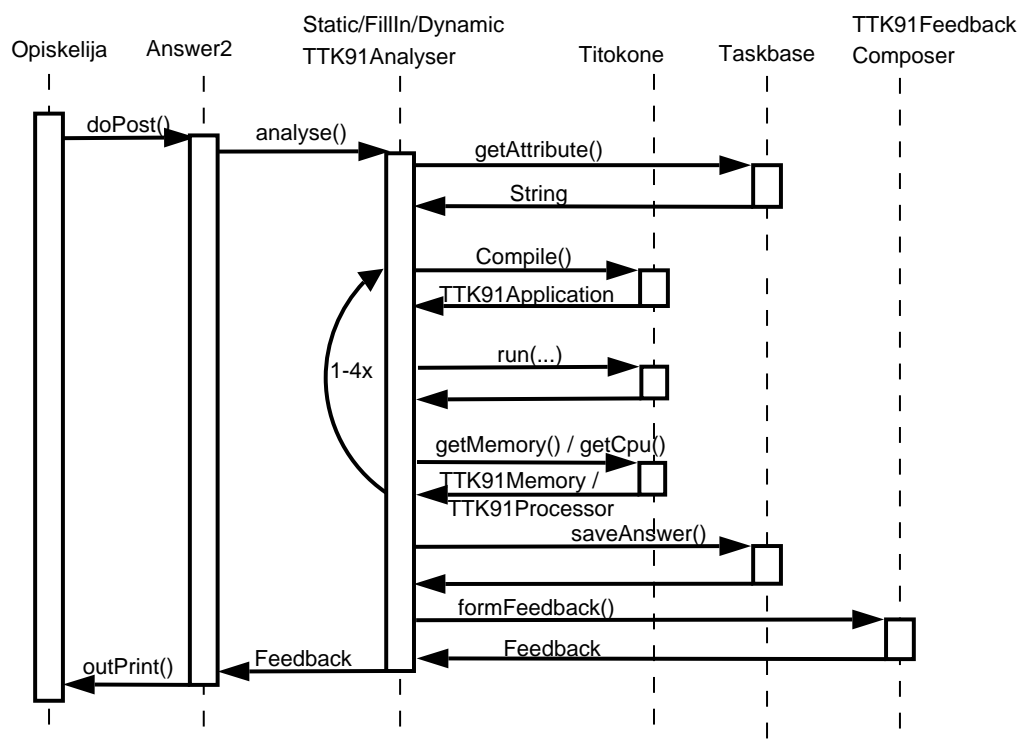


Kuva 4: Tehtävän esityksen ja tarkastuksen sekvenssikaavio

2.1.3 Analyser

Ulkoinen osajärjestelmä eAssari kutsuu Analyseria, kun opiskelija on tehnyt tehtävän ja painaa vastauslomakkeen lähetä-painiketta. Analyser ottaa opiskelijan ratkaisun vastaan ja tutkii sen oikeellisuutta ja laatua. Analyser vertaa ennalta määrättyjä tarkastuskriteerejä [Kr04a] opiskelijan syöttämään ratkaisuun ja Titokoneen tilaan ratkaisun simuloinnin jälkeen. Analyser käyttää ratkaisun arvioinnin apuna ulkoista osajärjestelmää Titokonetta. Jokaiselle tehtävätyypille on oma TTK91Analyser-luokka.

Titokoneen suorittaman simuloinnin tuloksena saadaan ohjelman tila suorituksen lopussa. Titokoneen tuloksen ja kriteerien perusteella muodostetaan palaute. Palaute muodostaa erillinen luokka TTK91FeedbackComposer. Palaute annetaan eAssarille, joka näyttää sen opiskelijalle. Analyserin toimintaa esittävä sekvenssikaavio on kuvassa 5.



Kuva 5: Tehtävän tarkastuksen sekvenssikaavio

2.2 Ulkoiset osajärjestelmät

Järjestelmä liittyy kahteen ulkoiseen osajärjestelmään. Ne ovat *Titokone* ja *eAssari*.

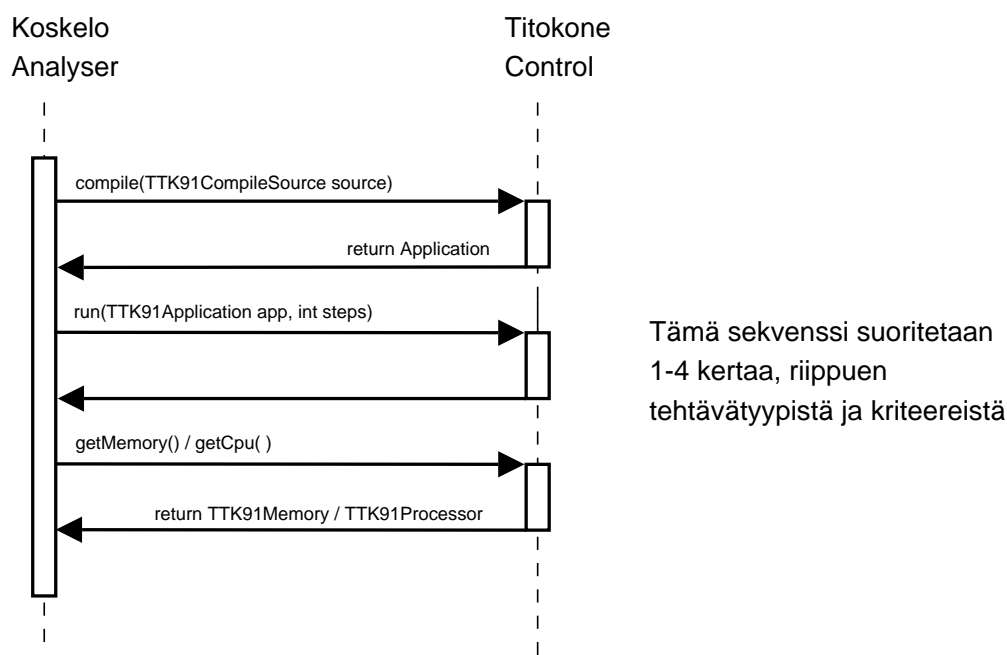
2.2.1 Titokone

Ulkoista järjestelmää Titokonetta käytetään opiskelijan luomien ratkaisujen tarkastamisen apuvälineenä. Sille syötetään TTK-91-kielinen ohjelma, joka käännetään ja ajetaan. Tiedot käännokestä ja suorituksesta välitetään Analyserille. Kuvassa 6 esitellään Analyserien ja Titokoneen yhteistyötä.

2.2.2 eAssari

Ulkoinen järjestelmä eAssari tarjoaa ympäristön, jossa opiskelija voi valita tehtävätyypin ja suoritettavan tehtävän. Valinnan jälkeen eAssari käynnistää Displayerin, joka muodostaa varsinaisen tehtävän. Tehtävän esittämisestä huolehtii eAssari.

Uutta tehtävää luotaessa eAssari käynnistää Composerin. Kun tehtävä on luotu, Composer antaa valmiin tehtävän eAssarille. Tämän jälkeen eAssari tallentaa teh-



Kuva 6: Analyserien ja Titokoneen yhteistyön sekvenssi

tävän tietokantaan.

Opiskelijan antama vastaus lähetetään ensin eAssarille, joka antaa sen Analyser-osajärjestelmälle tarkastettavaksi. Analyserin opiskelijan vastauksesta luoma Feedback-olio annetaan eAssarille, joka luo sen perusteella opiskelijan palautesivun.

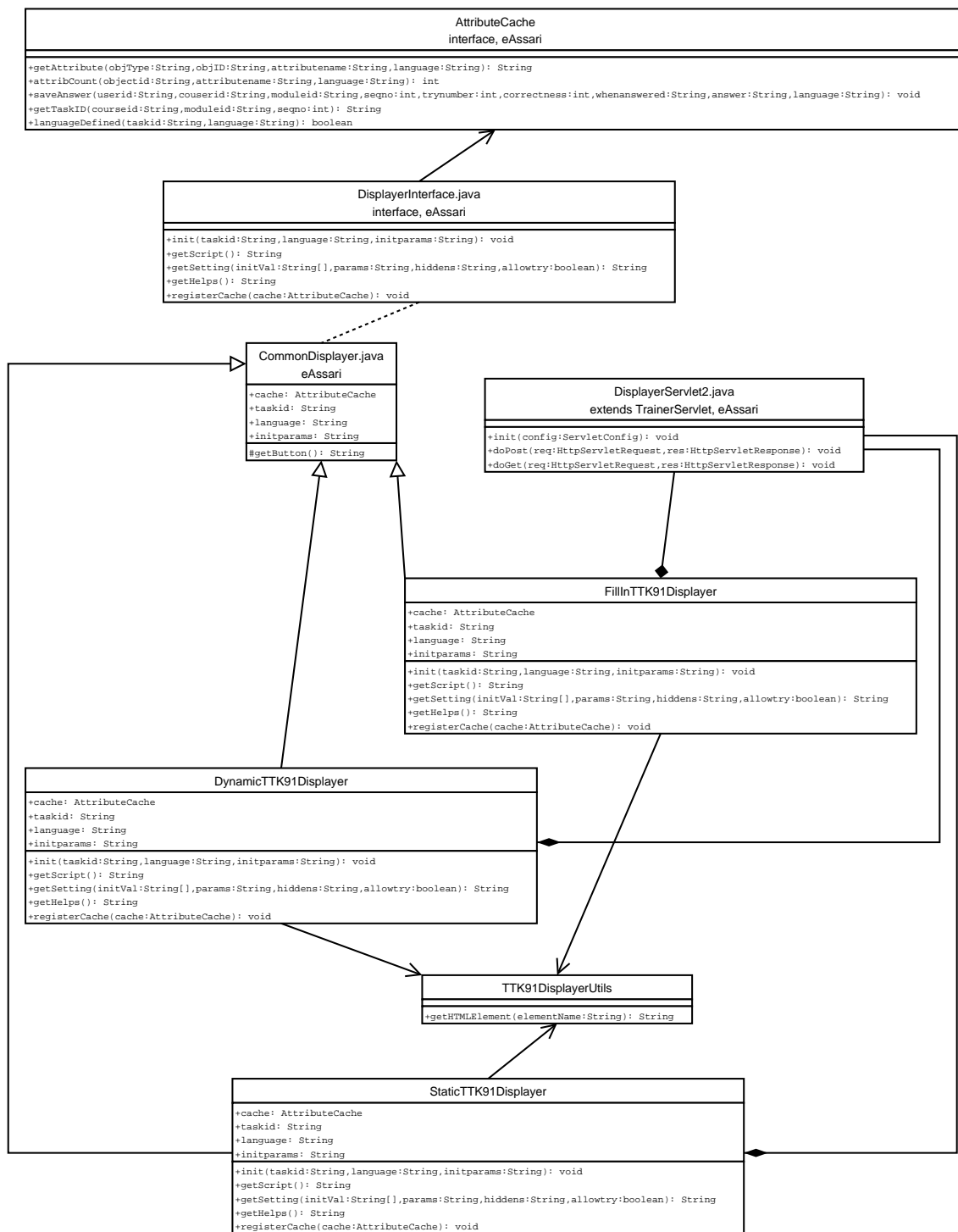
3 Luokkarakenne ja jsp-sivut

Järjestelmän luokkarakenne on kuvassa 7. Järjestelmä jakautuu kolmeen osaan, joista kukin kuvaa yhtä keskeistä osajärjestelmää. Ylimpänä on tehtävien tarkastamisen hoitava osajärjestelmä *Analyser*. Sitä seuraavat tehtävien luomisesta (*Composer*) ja näyttämisestä (*Displayer*) huolehtivat osajärjestelmät.

Osajärjestelmien tarkempi rakenne luokkineen ja metodikuvauksineen esitellään seuraavissa luvuissa.

3.1 Displayer-osajärjestelmä

Displayer-osajärjestelmän tehtävänä on luoda ja esittää käyttäjälle näytettävät html-sivut. Displayerin toimintaa esittävä luokkakaavio kuvassa 8.



Kuva 8: Displayer osajärjestelmä

3.1.1 CommonDisplayer implements DisplayerInterface

Abstrakti eAssarin-luokka, joka toteuttaa Displayer-rajapinnan. Metodien kuvaukset TTK91Displayerien alla.

```
public String init(String taskid, String language, String initparams)

public String getScript()

public void getSetting(String initVal[], String params, String hidden, boolean allowtry)

public String getHelps()

public void registerCache(AttributeCache cache)
```

3.1.2 DynamicTTK91Displayer extends CommonDisplayer

Arpoo dynaamisiin tehtäviin muuttuvien osien arvot, kun tehtävää kutsutaan ensimmäisen kerran. Kun arvot päätetty, Displayer tallentaa arvot kantaan TaskBasen avulla. Arvot tallennetaan kantaan, jotta Analyser pystyy myöhemmin päättelemään opiskelijan ratkaisun oikeellisuuden.

Luo valitun tehtävän tiedoista html-koodia, jonka palauttaa String-oliona. Osa html-koodista saadaan TTK91DisplayerUtils-luokan metodeilta.

```
public String init(String taskid, String language, String initparams)
Asettaa Displayerin kielen, tehtävän tunnisteiden sekä alkuarvot.
```

```
public String getScript()
Palauttaa sivulla käytettävän java-script koodin, jos sellaista on.
```

```
public void getSetting(String initVal[], String params, String hidden, boolean allowtry)
```

Palauttaa Displayerin koostaman html-sivun String-oliona.

```
public String getHelps()
```

```
public void registerCache(AttributeCache cache)
```

Antaa Displayerille viitteen AttributeCache-rajapinnan toteuttavasta luokasta. Parametri TaskBase-tyyppiä joka toteuttaa AttributeCachen. TaskBasella päästään käsiksi tietokantaan.

3.1.3 FillInTTK91Displayer extends CommonDisplayer

Luo valitun tehtävän tiedoista html-koodia, jonka palauttaa String-oliona. Osa html-koodista saadaan TTK91DisplayerUtils-luokan metodeilta.

`public String init(String taskid, String language, String initparams)`
Asettaa Displayerin kielen, tehtävän tunnisteiden sekä alkuarvot.

`public String getScript()` Palauttaa sivulla käytettävän java-script koodin, jos sellaista on.

`public void getSetting(String initVal[], String params, String hidden, boolean allowtry)`

Palauttaa Displayerin koostaman html-sivun String-oliona.

`public String getHelps()`

`public void registerCache(AttributeCache cache)`

Antaa Displayerille viitteen AttributeCache-rajapinnan toteuttavasta luokasta. Parametri TaskBase-tyyppiä joka toteuttaa AttributeCachen. TaskBasella päästään käsiksi tietokantaan.

3.1.4 StaticTTK91Displayer extends CommonDisplayer

Luo valitun tehtävän tiedoista html-koodia, jonka palauttaa String-oliona. Osa html-koodista saadaan TTK91DisplayerUtils-luokan metodeilta.

`public String init(String taskid, String language, String initparams)`
Asettaa Displayerin kielen, tehtävän tunnisteiden sekä alkuarvot.

`public String getScript()`
Palauttaa sivulla käytettävän java-script koodin, jos sellaista on.

`public void getSetting(String initVal[], String params, String hidden, boolean allowtry)`

Palauttaa Displayerin koostaman html-sivun String-oliona.

`public String getHelps()`

`public void registerCache(AttributeCache cache)`

Antaa Displayerille viitteen AttributeCache-rajapinnan toteuttavasta luokasta. Parametri TaskBase-tyyppiä joka toteuttaa AttributeCachen. TaskBasella päästään käsiksi tietokantaan.

3.1.5 TTK91DisplayerUtils

Sisältää apumetodeita Displayer-luokkia varten. Metodit palauttavat mm. valmiita html-elementtejä String-olioina, joita on helppo liittää toisiinsa kun muodostetaan sivua.

`public String getHTMLElement()` Näitä metodeja siis useita. Metodit päättää vasta luokan koodaaja koska elementtejä on vaikea määritellä etukäteen.

3.1.6 DisplayerServlet2

eAssarin servletti, joka tunnistaa valitun tehtävän, ja noutaa oikean Displayerin. Tulostaa html-sivun käyttäjälle käyttäen apuna Displayer-luokkia.

Oikean Displayerin noutaminen:

- TaskID saadaan TaskBasesta parametreillä courseid, moduleid ja seqno
- Task saadaan TaskBasesta TaskID:llä
- Taskista saadaan TaskType
- TaskTypestä saadaan Displayer

3.2 Composer-osajärjestelmä

Composer-osajärjestelmä tarjoaa palvelut uusien tehtävien luomiseen. Sen rakennetta valottava luokkakaavio kuvassa 9.

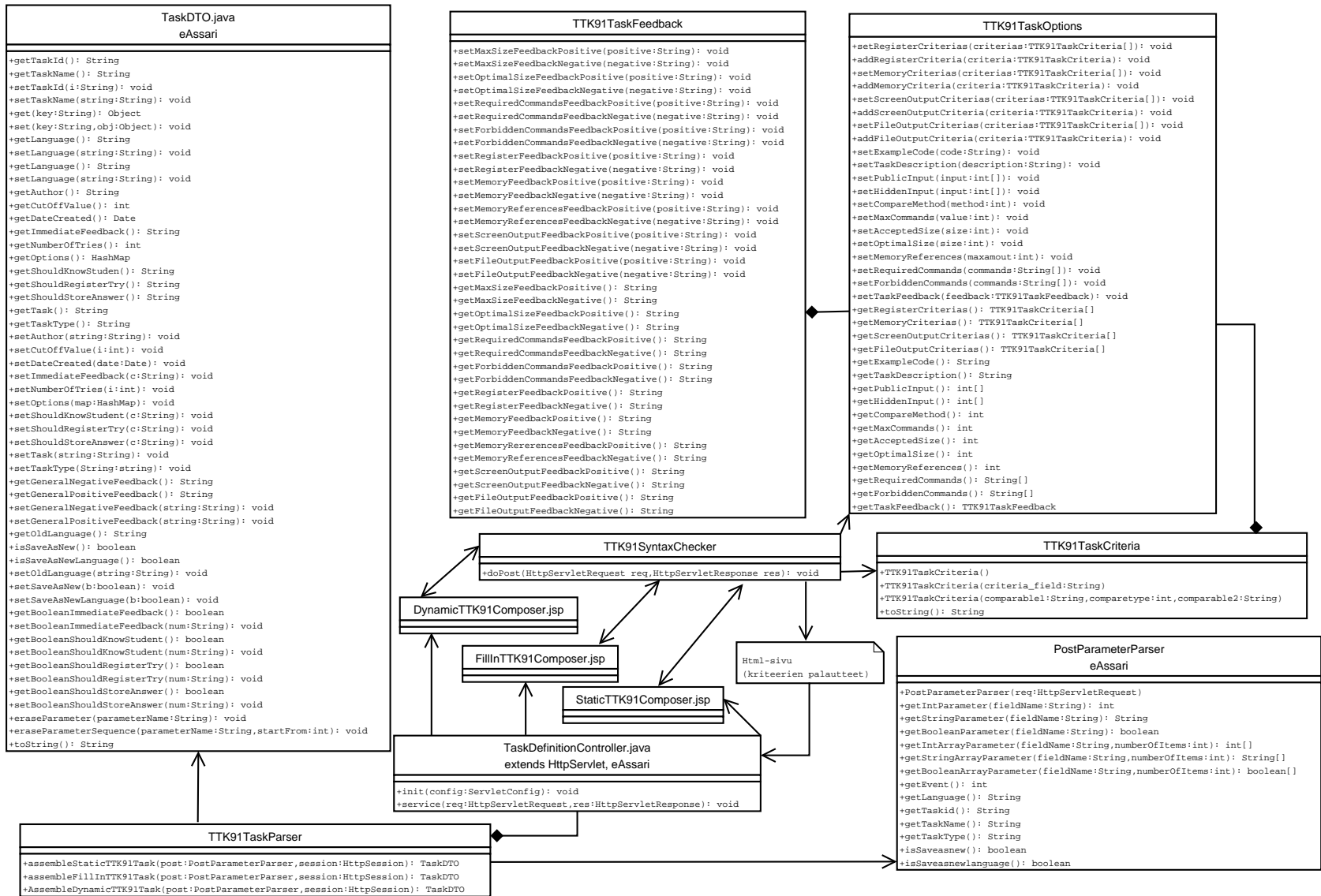
3.2.1 StaticTTK91Composer

JSP-sivu joka toteuttaa staattisen TTK91-tehtävämäärittelyn käyttöliittymän. Sivulla voi sekä luoda uuden tehtävän, että muuttaa vanhaa. Sivun kutsuu syntaksin tarkistamiseksi servlettiä SyntaxChecker, joka tarkistaa kunkin kriteerin syntaktisen oikeellisuuden. Sivulle palataan mikäli jokin kriteeri oli syntaksiltaan virheellinen ja annetaan käyttäjälle virheilmoitus, joka kertoo mikä kriteeri oli virheellinen.

StaticTTK91Composerin lomake sisältää seuraavat kentät jotka lähetetään SyntaxCheckerille:

- Malliratkaisu: exampleCode

Kuva 9: Composer-osajärjestelmän luokkakavio



- Tehtävänanto: taskDescription
- Opiskelijan syötteet: publicInput
- Salaiset syötteet: hiddenInput
- Vertailuperuste (malliratk. vai kiinteät arvot): compareMethod
- Silmukan esto: maxCommands
- Hyväksymisraja koolle (käskyjen määrä): acceptedSize
- Hyvä koko ohjelmalle: optimalSize
- Vaaditut käskyt: requiredCommands
- Kielletyt käskyt: forbiddenCommands
- Rekisterien arvot: registerValues
- Muistipaikkojen ja muuttujien arvot: memoryValues
- Tulosteet CRT:lle: screenOutput
- Tulosteet tiedostoon: fileOutput
- Muistiviitteiden yläraja: memoryReferences

3.2.2 DynamicTTK91Composer

Kuten staattisen tehtävän määrittelevä jsp-sivu, mutta tulostaa sivun, jossa on laajennuksia dynaamisten tehtävien tietojen täyttämiseksi. Laajennukset lisätään myöhemmin toteutusvaiheessa.

3.2.3 FillInTTK91Composer

Kuten staattisen tehtävänmäärittelyn jsp-sivu, mutta tarjoaa lisäkenttänä määrittelymahdollisuuden osalle TTK91-ohjelmakoodista. Laajennukset lisätään myöhemmin toteutusvaiheessa.

3.2.4 TTK91TaskParser

Apuluokka TaskDefinitionControlleria varten. Koostaa annetusta datasta TaskDTO-olion. TTK91Taskparser saa parametrinä PostParameterParserin sekä HttpSession.

PostParameterParser tarjoaa post-kutsun mukana tulleet palaute-kentät ja HttpSessiosta saadaan TTK91TaskOptions-olio joka sisältää kaiken muun tehtävän määrittelyyn.

Luokka saa tietää TTK91TaskOptionsin metodien avulla montako kriteerin palautetta PostParameterParser tarjoaa (tarkistamalla ko. kriteereiden taulukon koon esim. getMemoryCriteria.size()).

Postin mukana tulleet kentät on nimetty juoksevasti (ks. luokka SyntaxChecker) samalla tavalla. Kun TTK91TaskParser tietää palautteiden määrän, niin se voi liittää ne oikeisiin kriteereihin.

3.2.5 TTK91SyntaxChecker

Lukee jsp-sivulla määritellyn tehtävän. Tarkistaa, että kriteereiden syntaksi oikea, ja generoi käyttäjälle uuden sivun. Kriteereitä käsitellään TTK91Criteria-olioina. Kriteereiden syntaktinen oikeellisuus tarkistetaan uutta oliota luotaessa.

Jos jokin kriteeri ei ole syntaksiltaan oikein, SyntaxChecker palauttaa virheen. Kriteerit palautetaan takaisin TTK91Composerille, joka ilmoittaa tehtävänlaatijalle jonkin kriteerin olevan virheellinen.

Jos kriteerien syntaksi oli oikein, ohjataan käyttäjä uudelle sivulle. Tällä sivulle on jokaista kriteerikenttää kohden kaksi tekstikenttää palautetta varten. Toinen kenttä on negatiivista ja toinen positiivista palautetta varten koskien kriteerien täyttymistä.

Kentät on nimetty seuraavasti:

Positiiviset:	Negatiiviset:
registerFeedbackPositive	registerFeedbackNegative
memoryFeedbackPositive	memoryFeedbackNegative
screenOutputFeedbackPositive	screenOutputFeedbackNegative
fileOutputFeedbackPositive	fileOutputFeedbackNegative
requiredCommandsFeedbackPositive	requiredCommandsFeedbackNegative
forbiddenCommandFeedbackPositive	forbiddenCommandsFeedbackNegative
acceptedSizeFeedbackPositive	acceptedSizeFeedbackNegative
optimalSizeFeedbackPositive	optimalSizeFeedbackNegative
memoryReferencesFeedbackPositive	memoryReferencesFeedbackNegative

3.2.6 TTK91TaskCriteria

Kääreluokka joka sisältää yhden kriteerin.

```
public TTK91TaskCriteria(String criteriaField) Asettaa kriteerin ja tutkii
```

samalla sen syntaksin. Heittää poikkeuksen jos syntaksi väärin.

```
public TTK91TaskCriteria(String firstComparable, intComparator, String
secondComparable)
```

Asettaa kriteerin osiin jaettuna. Vertailun ensimmäisen muuttujan, vertailuoperaation ja toisen vertailun muuttujan.

```
public String toString()
```

Palauttaa kriteerin merkkiesityksenä.

3.2.7 TTK91TaskFeedback

Kääreluokka joka sisältää kaiken tehtävään liittyvän palautteen.

```
public void setAcceptedSizeFeedbackPositive(String positive)
```

Asettaa positiivisen palautteen ohjelman maksimikoosta.

```
public void setAcceptedSizeFeedbackNegative(String negative)
```

Asettaa negatiivisen palautteen ohjelman maksimikoosta.

```
public void setOptimalSizeFeedbackPositive(String positive)
```

Asettaa positiivisen palautteen ohjelman optimaalisesta koosta.

```
public void setOptimalSizeFeedbackNegative(String negative)
```

Asettaa negatiivisen palautteen ohjelman optimaalisesta koosta.

```
public void setRequiredCommandsFeedbackPositive(String positive)
```

Asettaa positiivisen palautteen ohjelman vaadituista käskyistä.

```
public void setRequiredCommandsFeedbackNegative(String negative)
```

Asettaa negatiivisen palautteen ohjelman vaadituista käskyistä.

```
public void setForbiddenCommandsFeedbackPositive(String positive)
```

Asettaa positiivisen palautteen ohjelman kielletyistä käskyistä.

```
public void setForbiddenCommandsFeedbackNegative(String negative)
```

Asettaa negatiivisen palautteen ohjelman kielletyistä käskyistä.

```
public void setRegisterFeedbackPositive(String positive)
```

Asettaa positiivisen palautteen rekistereistä.

```
public void setRegisterFeedbackNegative(String negative)
```

Asettaa negatiivisen palautteen rekistereistä.

```
public void setMemoryFeedbackPositive(String positive)
```

Asettaa positiivisen palautteen muistipaikoista ja muuttujista.

`public void setMemoryFeedbackNegative(String negative)`
Asettaa negatiivisen palautteen muistipaikoista ja muuttujista.

`public void setMemoryReferencesFeedbackPositive(String positive)`
Asettaa positiivisen palautteen muistiviitteiden määrästä.

`public void setMemoryReferencesFeedbackNegative(String negative)`
Asettaa negatiivisen palautteen muistiviitteiden määrästä.

`public void setScreenOutputFeedbackPositive(String positive)`
Asettaa positiivisen palautteen näytön tulosteista.

`public void setScreenOutputFeedbackNegative(String negative)`
Asettaa negatiivisen palautteen näytön tulosteista.

`public void setFileOutputFeedbackPositive(String positive)`
Asettaa positiivisen palautteen tiedoston tulosteista.

`public void setFileOutputFeedbackNegative(String negative)`
Asettaa negatiivisen palautteen tiedoston tulosteista.

`public String getAcceptedSizeFeedbackPositive()`
Antaa positiivisen palautteen maksimikoosta.

`public String getAcceptedSizeFeedbackNegative()`
Antaa negatiivisen palautteen maksimikoosta.

`public String getOptimalSizeFeedbackPositive()`
Antaa positiivisen palautteen optimaalisesta koosta.

`public String getOptimalSizeFeedbackNegative()`
Antaa negatiivisen palautteen optimaalisesta koosta.

`public String getRequiredCommandsFeedbackPositive()`
Antaa positiivisen palautteen vaadituista käskyistä.

`public String getRequiredCommandsFeedbackNegative()`
Antaa negatiivisen palautteen vaadituista käskyistä.

`public String getForbiddenCommandsFeedbackPositive()`
Antaa positiivisen palautteen kielletyistä käskyistä.

`public String getForbiddenCommandsFeedbackNegative()`
Antaa negatiivisen palautteen kielletyistä käskyistä.

`public String getRegisterFeedbackPositive()`
Antaa positiivisen palautteen rekistereistä.

```
public String getRegisterFeedbackNegative()
```

Antaa negatiivisen palautteen rekistereistä.

```
public String getMemoryFeedbackPositive()
```

Antaa positiivisen palautteen muistipaikoista ja muuttujista.

```
public String getMemoryFeedbackNegative()
```

Antaa negatiivisen palautteen muistipaikoista ja muuttujista.

```
public String getMemoryReferencesFeedbackPositive()
```

Antaa positiivisen palautteen muistiviitteiden määrästä.

```
public String getMemoryReferencesFeedbackNegative()
```

Antaa negatiivisen palautteen muistiviitteiden määrästä.

```
public String getScreenOutputFeedbackPositive()
```

Antaa positiivisen palautteen näytön tulosteista.

```
public String getScreenOutputFeedbackNegative()
```

Antaa negatiivisen palautteen näytön tulosteista.

```
public String getFileOutputFeedbackPositive()
```

Antaa positiivisen palautteen tiedoston tulosteista.

```
public String getFileOutputFeedbackNegative()
```

Antaa negatiivisen palautteen tiedoston tulosteista.

3.2.8 TTK91TaskOptions

Kääreluokka joka sisältää tehtävän kaikki määrittelyt ja kriteerit.

```
public void addRegisterCriteria(criteria TTK91TaskCriteria)
```

Lisää rekisterin kriteerin rekisteri-taulukon viimeiseksi.

```
public void addRegisterCriterias(criterias [] TTK91TaskCriteria)
```

Lisää taulukkona annetut kriteerit entisten perään.

```
public void addMemoryCriteria(criteria TTK91TaskCriteria)
```

Lisää muistipaikan tai muuttujan kriteerin muisti-taulukon viimeiseksi.

```
public void addMemoryCriterias(criterias [] TTK91TaskCriteria)
```

Lisää taulukkona annetut kriteerit entisten perään.

```
public void addScreenOutputCriteria(criteria TTK91TaskCriteria)
```

Lisää näytön tulosteeseen liittyvän kriteerin crt-taulukon viimeiseksi.

```
public void addScreenOutputCriteria(criteria[] TTK91TaskCriteria)
Lisää taulukkona annetut kriteerit entisten perään.
```

```
public void addFileOutputCriteria(criteria TTK91TaskCriteria)
Lisää tiedostoon tulostamiseen liittyvän kriteerin file-aulukon viimeiseksi.
```

```
public void addFileOutputCriteria(criteria[] TTK91TaskCriteria)
Lisää taulukkona annetut kriteerit entisten perään.
```

```
public void setExampleCode(String code)
Asettaa mallivastauksen.
```

```
public void setTaskDescription(String description)
Asettaa tehtävänannon.
```

```
public void setPublicInput(int[] input)
Asettaa public-aulukkoon opiskelijalle näytettävät syötteen.
```

```
public void setHiddenInput(int[] input)
Asettaa hidden-aulukkoon salaiset testisyötteen.
```

```
public void setCompareMethod(int method)
Asettaa arvon miten tehtävän oikeellisuus tutkitaan. Arvolla 0 käytetään mallivastausta. Arvolla 1 käytetään annettuja arvoja.
```

```
public void setMaxCommands(int value)
Asettaa rajan suoritettavien komentojen määrälle.
```

```
public void setAcceptedSize(int size)
Asettaa hyväksymisrajan ohjelman koolle käskyissä mitattuna.
```

```
public void setOptimalSize(int size)
Asettaa ohjelman ihannekoon käskyissä mitattuna.
```

```
public void setRequiredCommands(String[] commands)
Asettaa ohjelmassa vaaditut komennot.
```

```
public void setForbiddenCommands(String[] commands)
Asettaa ohjelmassa ehdottomasti kielletyt komennot.
```

```
public TTK91TaskCriteria[] getRegisterCriteria()
Hakee kaikki rekistereihin liittyvät kriteerit.
```

```
public TTK91TaskCriteria[] getMemoryCriteria()
Hakee kaikki muistiin liittyvät kriteerit.
```

```
public TTK91TaskCriteria[] getScreenOutputCriteria()
Hakee kaikki näytön tulosteisiin liittyvät kriteerit.
```



```
public TTK91TaskCriteria[] getFileOutputCriteriaas()
```

Hakee kaikki tiedoston tulosteisiin liittyvät kriteerit.

```
public String getExampleCode()
```

Hakee mallivastauksen.

```
public String getTaskDescription()
```

Hakee tehtävänannon.

```
public int[] getPublicInput()
```

Hakee opiskelijalle näytettävät syötteet.

```
public int[] getHiddenInput()
```

Hakee salaiset testisyötteet.

```
public int getCompareMethod()
```

Hakee vertailutavan jolla tehtävä tarkastetaan.

```
public int getMaxCommands()
```

Hakee luvun, joka ilmoittaa kuinka monta komentoa korkeintaan ajetaan Titokoneella.

```
public getAcceptedSize()
```

Hakee luvun, joka ilmoittaa kuinka monta komentoa koodissa saa enintään olla.

```
public int getOptimalSize()
```

Hakee luvun, joka ilmoittaa mikä olisi ohjelman komentojen ihannemäärä.

```
public String[] getRequiredCommands()
```

Hakee ohjelmassa vaaditut komennot.

```
public String[] getForbiddenCommands()
```

Hakee ohjelmassa kielletyt komennot.

3.2.9 TaskDefinitionController

Assarin servletti, joka ohjaa tehtävän luontiin liittyvät kutsut oikeille metodeille tai jsp-sivuille kutsukoodien perusteella.

Servletti saa parametrina kokonaisluvun, joka kuvaa tapahtumaa. Tapahtumasta riippuen servletti käynnistää muita palveluita tai kutsuu ulkoisia osajärjestelmiä. Kaikki tehtävänmäärittelyyn liittyvät tapahtumat kulkevat tämän servletin kautta.

Myös TaskDefinitionController on servletti ja sen kautta eAssari käynnistää oikean tehtävänmäärittelysivun. Tehtävänmäärittelysivut (eri Composerit) on toteutettu jsp-sivuina.

3.3 Analyser-osajärjestelmä

Analyser-osajärjestelmä toimii kiinteästi yhteistyössä ulkoisen osajärjestelmän, Titokoneen kanssa. Se tutkii opiskelijan ratkaisuyritysten oikeellisuutta, ja antaa niihin liittyvän palautteen. Analyserin rakennetta kuvaava luokkakaavio kuvassa 10.

3.3.1 Answer2

Answer2 on eAssarin servletti, joka ottaa vastaan opiskelijan ratkaisun ja kutsuu tehtävätyyppiin liittyvää Analyseria. Oikea Analyser saadaan kutsumalla Task-luokan getAnalyser-metodia. Analyysin jälkeen servletti luo opiskelijan palautesivun. Luokan toiminta kuvassa 11.

3.3.2 CommonAnalyser implements AnalyserInterface

Abstrakti eAssarin luokka, joka on kaikkien TTK91Analyser-luokkien ylliluokka. CommonAnalyser-luokka toteuttaa AnalyserInterface-rajapinnan. Metodien kuvaukset ovat StaticTTK91Analyserin alla.

```
abstract public void init(String taskid, String language, String init-
params)
```

```
abstract public Feedback analyse(String[] answer, String params)
```

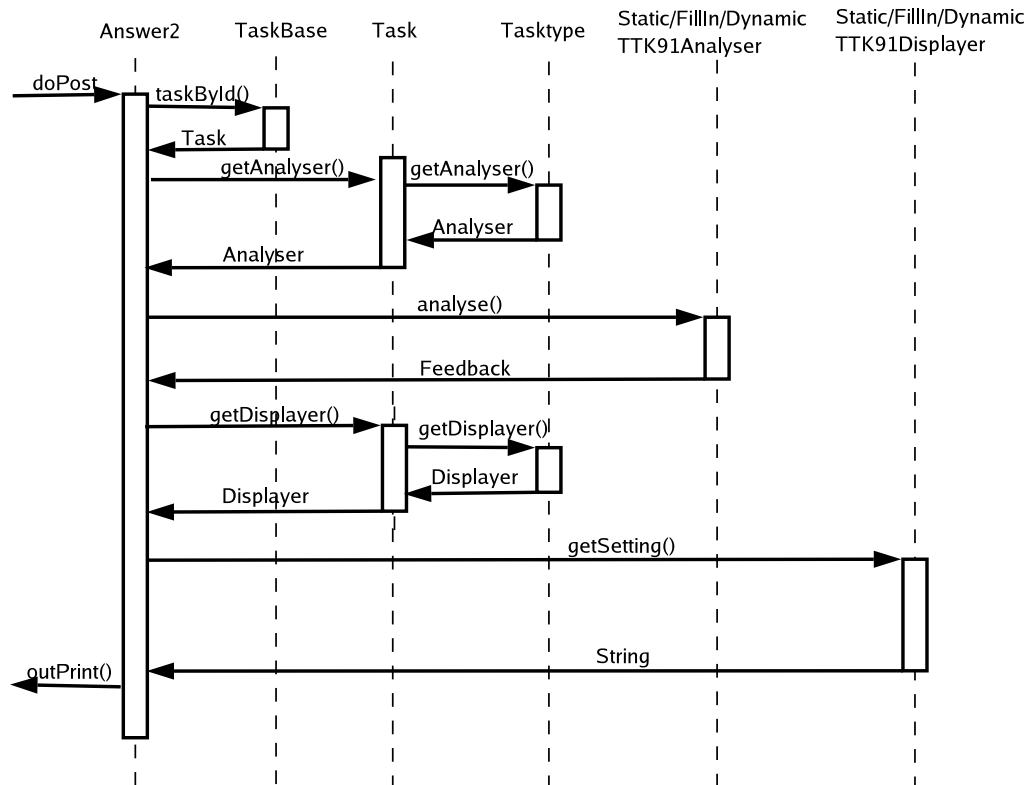
```
abstract public void registerCache(AttributeCache c)
```

3.3.3 StaticTTK91Analyser extends CommonAnalyser

Kutsutaan Answer2-servletistä. StaticTTK91Analyser arvioi staattisen tehtävän ratkaisun Titokonetta avuksi käyttäen. Titokoneen suorittaman simuloinnin tuloksena saadaan ohjelman tila suorituksen lopussa.

Simulointeja voidaan joutua tekemään 1-4 kertaa (kuva 5, sivulla 6). Simulointien määrä riippuu siitä käytetäänkö mallivastausta arviointiin ja annetaanko ohjelmille vaihtoehtoisia syötteitä. TTK91-ohjelman suoritus tapahtuu Control-luokan kautta. Lopuksi StaticTTK91Analyser antaa analyysin tuloksen TTK91FeedbackComposerille, joka muodostaa Feedback-olion.

```
public void init(String taskid, String language, String initparams)
Alustaa StaticTTK91Analyserin tehtävätunnuksella, kielellä ja alustusparametreil-
la.
```

Kuva 11: Answer2-luokan toiminta

```
public Feedback analyse(String[] answer, String params)
```

Analysoi ratkaisun käyttämällä Titokonetta. Saa parametreinä tehtävän vastauksen String-taulukkona ja valinnaiset parametrit String-oliona.

```
public void registerCache(AttributeCache c)
```

Antaa viitteen AttributeCache-rajapinnan toteuttavaan TaskBase-luokkaan. Tämän jälkeen voidaan hakea tehtävään kuuluvia attribuutteja getAttribute-metodilla.

3.3.4 FillInTTK91Analyser extends CommonAnalyser

Toimii muuten kuten StaticTTK91Analyser, mutta arvioi täydennystehtävän ratkaisun. Sisältää samat metodit kuin StaticTTK91Analyser.

3.3.5 DynamicTTK91Analyser extends CommonAnalyser

Toimii muuten kuten StaticTTK91Analyser, mutta arvioi dynaamisen tehtävän ratkaisun. Sisältää samat metodit kuin StaticTTK91Analyser.

3.3.6 TTK91FeedbackComposer

Kutsutaan TTK91Analyser-luokista. FeedbackComposerin tehtävä on muodostaa palautteesta Feedback-olio.

```
public Feedback formFeedback(TTK91Criteria[] criteria, Task task)
```

Metodi saa parametrinaan kaikki tehtävän tarkastamisessa käytettävät kriteerit criteria taulukossa. Luo kriteerien perusteella Feedback-olion.

3.3.7 TTK91AnalyserUtils

Kutsutaan kaikista TTK91Analyser-luokista. Luokka sisältää kaikille tehtävätyypeille yhteisten kriteerien arviointimetodeja. Metodit päättää vasta luokan koodaaja.

3.3.8 TaskBase implements AttributeCache

Taskbase on eAssarin luokka ja se toimii rajapintana tehtävätietokantaan. Luokan ilmentymä säilyttää tehtäviä tietoineen omassa keskusmuistitietorakenteessaan ja hoitaa tarvittavan kommunikoinnin tietokannan kanssa muun muassa tallentamalla sinne tehtävän ratkaisuyritykset.

3.3.9 Control (Titokoneen rajapintaluokka)

Kuvassa 6 (sivulla 7) Titokoneen Control-rajapintaluokan ulkoinen toiminta. Tehtäväkohtaiset Analyserit kutsuvat Titokoneen Control-rajapintaluokan metodeja. Ensin pyydetään käännöspalvelua, joka palauttaa ajettavan TTK91-ohjelman. Tämä ohjelma laitetaan suoritukseen, jonka jälkeen suorituksesta pyydetään tiedot (vrt.5.2).

```
public TTK91Application compile(TTK91CompileSource source)
```

Metodi joka palauttaa Titokoneella ajettavan TTK91-ohjelman.

```
public void run(TTK91Application app, int steps)
```

Run-metodille annetaan parametriksi käännetty TTK91-ohjelma, ja ajettavien konekäskyjen maksimimäärä.

```
public TTK91CPU getCPU()
```

Palauttaa ilmentymän Titokoneen Control-luokasta.

```
public TTK91Memory getMemory()
```

Palauttaa ilmentymän Titokoneen RancamAccessMemory-luokasta.

4 Käyttöliittymä

Järjestelmän käyttöliittymä muodostuu html-sivuista. Sivut ovat käyttäjän näkökulmasta staattisia, normaaleja html-sivuja. Ne sisältävät normaaleja html-elementtejä kuten lomakkeita (form) ja tekstikappaleita.

Esimerkiksi opiskelijan yritettyä tehtävän ratkaisua, muodostaa Answer2 palautteosan sivusta Analyser-osajärjestelmän tarjoaman Feedback-olion avulla. Tällöin Analyser muodostaa Feedback-olion, Displayer tehtävänannon html-esityksen ja Answer2 kokoaa sivun valmiiksi.

Ohjelmiston ulkoisille toiminnoille (tehtävän luonti ja muokkaus, tehtävän ratkaiseminen) on omat käyttöliittymänsä.

Käyttöliittymähahmotelmat kuvissa 12 ja 13.

4.1 Composer – tehtävän luonti ja muokkaus

Composerin käyttöliittymä koostuu html-kaavakkeesta, johon syötetään tehtävänannon lisäksi myös esimerkiksi kriteerimäärittelykset.

4.1.1 Kriteerien syntaksi

Käskyjä, rekistereitä, muistipaikkoja, muuttujia ja tulosteita määriteltäessä kaikki kriteerit kentissä tulee erottaa toisistaan puolipisteellä, sekä erotuksen lisäksi kriteeri on laitettava sulkuihin. Esim:

```
(R1 = 100);
(R3 < 50);
(R6 != 10);
```

Mikali kriteeri on laadullinen, erotetaan siitä kertova 'L' pilkulla. Esim:

```
(L,R1 = 100);
```

Jos määritellään samalla kertaa useampi arvo, erotetaan arvot toisistaan pilkuilla. Esimerkiksi tulosteiden määrittelyssä:

```
(1,50);
(3,30);
```

Diskreettejä joukkoja määriteltäessä joukko on laitettava sulkeisiin, ja arvot erotel-

Tehtävänanto:

Opiskelijalle näytettävät syötteen:

Salaiset syötteen:

Malliratkaisu Vertailu: Valmiit kriteerit
Malliratkaisu

Suoritettavien käskyjen maksimimäärä:

Hyväksyttävän ratkaisun käskyjen maksimimäärä:

Ohjelman ihannekoko (käskyjä):

Ohjelmassa vaaditut käskyt

Ohjelmassa kielletyt käskyt

Rekisterien sisältö

Muistipaikkojen ja muuttujien sisältö

Muistiviitteiden määrä

Tulosteet näytölle

Tulosteet tiedostoon

Tehtävän luonti
osa 2

Kuva 12: Käyttöliittymähahmotelma: staattisen tehtävän luonti

Opiskelijalle annettavat palautteet

Kriteeri	Kriteeri täyttyy	Kriteeri ei täyty
Hyväksyttävän ratkaisun käskyjen maksimimäärä	<input type="text"/>	<input type="text"/>
Ihannekoko	<input type="text"/>	<input type="text"/>
Ohjelmassa vaaditut käskyt	<input type="text"/>	<input type="text"/>
Ohjelmassa kielletyt käskyt	<input type="text"/>	<input type="text"/>
Rekisterien sisältö	<input type="text"/>	<input type="text"/>
Muistipaikkojen ja muuttujien sisältö	<input type="text"/>	<input type="text"/>
Muistiviitteiden määrä	<input type="text"/>	<input type="text"/>
Tulosteet näytölle	<input type="text"/>	<input type="text"/>
Tulosteet tiedostoon	<input type="text"/>	<input type="text"/>

Luo ja tallenna
tehtävä

Kuva 13: Käyttöliittymähahmotelma: palautemäärittely

tava pilkulla. Loppuun ei tarvitse puolipistettä. Esim:

(1,2,3,4)

4.1.2 Symbolien merkitykset kriteerimäärittelyssä

Symboli	Merkitys
RX	Viitattu rekisteri R jossa X on kokonaisluku
L	Kriteeri on laadullinen

FIXME: taulukko tähän ;)

4.1.3 Dynaamisen tehtävän määrittelyn syntaksi

Dynaamisen tehtävän määrittelyn syntaksi noudattaa em. annettua. Dynaamisten osien ilmaisussa noudatetaan kuitenkin lisäksi seuraavaa:

#J viittaa diskreettiin joukkoon

#A viittaa arvoväliin

#M viittaa muuttujien joukkoon

#R viittaa rekisterien joukkoon

Dynaamisessa tehtävänannossa ja koodissa ilmaistaan em. tageilla viite muuttuvaan osaan. Tehtävää generoitaessa, tehtävänannosta ja mallikoodista vaihdetaan tagit arvoiksi, jotka Displayer on arponut ko. joukosta tai arvovälistä.

Toisinaan on tarpeen määrittellä atomisia osia mallikoodista. Tämä tapahtuu ympäröimällä atomiset osat koodista hakasuluilla. Esimerkiksi:

```
LOAD, R1 = 100
[ADD, R2 = 10]
[ADD, R2 = 20]
SVC, = HALT
```

Jos koodin osa on merkattu atomiseksi voidaan se helposti yhdistää diskreettiin joukkoon. Jokainen diskreetin joukon alkio vastaa järjestyksessä yhtä koodin atomista osaa.

Eli kun Displayer on arponut käytettävän joukon, niin se tallentaa tehtävätiedot opiskelijan tiedoilla kantaan. Myöhemmin Analyser voi hakea talletetut tiedot kannasta, ja luoda oikean malliratkaisun poistamalla koodista tarpeettomia osia.

Eli mitään ei siis generoida, vaan määrittelijän koodista poistetaan käyttämätön osa jolloin saadaan mallivastaus.

4.2 Displayer – tehtävän ratkaisu ja palaute

4.3 Analyser – vastauksen analysointi

Analyserilla ei ole omaa, käyttäjälle näkyvää käyttöliittymää. Tarkastettuaan tehtävän se muodostaa palautteen ratkaisusta, ja tarjoilee sen Feedback-oliona Answer2:lle.

5 Muutokset ja lisäykset ulkoisiin osajärjestelmiin

Tässä luvussa käsitellään muutokset ja lisäykset, jotka tehdään ulkoisiin osajärjestelmiin.

5.1 Lisäykset eAssariin

eAssarin Events-luokkaan lisätty seuraavat tehtävätyyppien tunnisteet:

`STATIC_TTK91_COMPOSE = 27`

Halutaan näyttää käyttäjälle staattisen tehtävämäärittelyn sivu uutta tehtävää varten.

`STATIC_TTK91_EDIT = 28`

Halutaan editoida jo olemassaolevaa staattista tehtävää.

`STATIC_TTK91_SUBMIT = 29`

Halutaan tallentaa määritelty tehtävä.

`FILLIN_TTK91_COMPOSE = 30`

Halutaan näyttää käyttäjälle täydennystehtävän määrittelysivu uutta tehtävää varten.

`FILLIN_TTK91_EDIT = 31`

Halutaan editoida jo olemassaolevaa täydennystehtävää.

`FILLIN_TTK91_SUBMIT = 32`

Halutaan tallentaa määritelty täydennystehtävä.

DYNAMIC_TTK91_COMPOSE = 33

Halutaan näyttää käyttäjälle dynaamisen tehtävän määrittelysivu uutta tehtävää varten.

DYNAMIC_TTK91_EDIT = 34

Halutaan editoida jo olemassaolevaa dynaamista tehtävää.

DYNAMIC_TTK91_SUBMIT = 35

Halutaan tallentaa määritelty dynaaminen tehtävä.

Luokkaan TaskTypes on määritelty uudet TTK91-tehtävät seuraavilla nimillä:

staticttk91task Staattinen ohjelmointitehtävä.

fillinttk91task Täydennystehtävä.

dynamicttk91task Dynaaminen ohjelmointitehtävä.

Uudet tehtävätyypit on lisätty kantaan seuraavasti:

```
insert into tasktype values ('staticttk91task', 'Koskelo-projektiryhmä',
to_date('05.11.2004', 'DD.MM.YYYY'), null,
'fi.helsinki.cs.koskelo.displayer.StaticTTK91Displayer',
'fi.helsinki.cs.koskelo.analyser.StaticTTK91Analyser',
'http://www.cs.helsinki.fi/group/assari/css/assari.css', null, null);
```

```
insert into tasktype values ('fillinttk91task', 'Koskelo-projektiryhmä',
to_date('05.11.2004', 'DD.MM.YYYY'), null,
'fi.helsinki.cs.koskelo.displayer.FillInTTK91Displayer',
'fi.helsinki.cs.koskelo.analyser.FillInTTK91Analyser',
'http://www.cs.helsinki.fi/group/assari/css/assari.css', null, null);
```

```
insert into tasktype values ('dynamicttk91task', 'Koskelo-projektiryhmä',
to_date('05.11.2004', 'DD.MM.YYYY'), null,
'fi.helsinki.cs.koskelo.displayer.DynamicTTK91Displayer',
'fi.helsinki.cs.koskelo.analyser.DynamicTTK91Analyser',
'http://www.cs.helsinki.fi/group/assari/css/assari.css', null, null);
```

Uusien tehtävätyyppien attribuutit on lisätty kantaan seuraavasti:

```
insert into taskattributes values ('staticttk91task', 'exampleCode', 'TEXT', 'N',
'STORED');
```

```
insert into taskattributes values ('staticttk91task', 'taskDescription', 'TEXT', 'Y',
STORED');
```

```
insert into taskattributes values ('statictk91task', 'publicInput', 'TEXT', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'hiddenInput', 'TEXT', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'compareMethod', 'NUM', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'maxCommands', 'NUM', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'acceptedSize', 'NUM', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'optimalSize', 'NUM', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'memoryReferences', 'NUM', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'requiredCommands', 'TEXT', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'forbiddenCommands', 'TEXT', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'registerValues', 'TEXT', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'memoryValues', 'TEXT', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'screenOutput', 'TEXT', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'fileOutput', 'TEXT', 'N', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'acceptedSizeFeedbackPositive', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'acceptedSizeFeedbackNegative', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'optimalSizeFeedbackPositive', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'optimalSizeFeedbackNegative',
'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'memoryReferencesFeedbackPo-
sitive', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'memoryReferencesFeedbackNe-
gative', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'requiredCommandsFeedbackPo-
sitive', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'requiredCommandsFeedbackNe-
gative', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'forbiddenCommandsFeedback-
Positive', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'forbiddenCommandsFeedback-
Negative', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'registerFeedbackPositive', 'TEXT',
'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'registerFeedbackNegative',
'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'memoryFeedbackPositive', 'TEXT',
'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'memoryFeedbackNegative',
'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'screenOutputFeedbackPositive',
'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'screenOutputFeedbackNegative',
'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'fileOutputFeedbackPositive',
'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'fileOutputFeedbackNegative',
'TEXT', 'Y', 'STORED');
```

Sivujen kieliversiot lisätty kantaan seuraavasti:

```
insert into attributevalues values ('D', 'statictk91taskcomposer', 'exampleCodeHea-
```

```
der', 'FI', 'C', 'Ohjelman malliratkaisu');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'taskDescription-Header', 'FI', 'C', 'Tehtävänanto');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'publicInputHeader', 'FI', 'C', 'Opiskelijalle näytettävät syötteen');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'hiddenInputHeader', 'FI', 'C', 'Salaiset syötteen');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'compareMethod1Header', 'FI', 'C', 'Vertailu malliratkaisun suorituksen perusteella');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'compareMethod2Header', 'FI', 'C', 'Vertailu valmiisiin kriteereihin');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'maxCommandsHeader', 'FI', 'C', 'Ohjelman käskyjen yläraja (silmukan esto)');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'acceptedSizeHeader', 'FI', 'C', 'Hyväksymisen yläraja');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'optimalSizeHeader', 'FI', 'C', 'Ohjelman ihannekoon yläraja');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'memoryReferencesHeader', 'FI', 'C', 'Muistiviitteiden määrä');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'requiredCommandsHeader', 'FI', 'C', 'Ohjelmassa vaaditut käskyt');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'forbiddenCommandsHeader', 'FI', 'C', 'Ohjelmassa kielletyt käskyt');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'registerValuesHeader', 'FI', 'C', 'Rekisterien sisältö');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'memoryValuesHeader', 'FI', 'C', 'Muistipaikkojen ja muuttujien sisältö');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'screenOutputHeader', 'FI', 'C', 'Tulosteet näytölle');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'fileOutputHeader', 'FI', 'C', 'Tulosteet tiedostoon');
```

```
insert into attributevalues values ('D', 'staticttk91taskcomposer', 'submitButton-
```

Text', 'FI', 'C', 'Tehtävän luonti osa 2');

5.2 Lisäykset Titokoneeseen

Kaikki muutokset on mainittu koodissa kommentteilla *"added by HT, pvm, koskelo-projekti"*.

RandomAccessMemory

Muuttujat:

private int memory_references

Metodit:

Metodi public int getMemoryReferences()

Palauttaa kokonaislukuna suoritettujen muistiviittausten määrän.

Processor

Muuttujat:

private int stack_size

private int commands_executed

Metodit:

public int giveCommAmount()

Palauttaa suoritettujen komentojen määrän.

public int giveStackSize()

Palauttaa pinon suurimman koon.

Lähteet

- Ar04 Assari-ryhmä, Assari-projektin kotisivut., 2004. <http://www.cs.helsinki.fi/group/assari/>. [3.11.2004]
- FSF99 Free Software Foundation, GNU Lesser General Public License, 1999. <http://www.gnu.org/copyleft/lesser.html>. [3.11.2004]
- Häk91 Häkkinen, A., Yhteenveto TTK-91 käskyistä, 1991. <http://www.cs.helsinki.fi/u/ahakkine/Tito/koksi.kaskyt>. [3.11.2004]
- Kr04a Koskelo-ryhmä, Koskelo-projektin vaatimusdokumentti., 2004. <http://www.cs.helsinki.fi/group/koskelo/>. [5.10.2004]
- Kr04b Koski-ryhmä, Koski-projektin kotisivut., 2004. <http://www.cs.helsinki.fi/group/koski/>. [3.11.2004]