

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
1.2	15.12.2004	Toteutusvaiheessa päivitetty versio

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	7
2.2.1	Titokone	7
2.2.2	eAssari	7
3	Luokkarakenne ja jsp-sivut	8
3.1	Displayer-osajärjestelmä	8
3.1.1	CommonDisplayer implements DisplayerInterface	8
3.1.2	DynamicTTK91Displayer extends CommonDisplayer	11
3.1.3	FillInTTK91Displayer extends CommonDisplayer	11
3.1.4	StaticTTK91Displayer extends CommonDisplayer	12
3.1.5	TTK91DisplayerUtils	12
3.1.6	DisplayerServlet2	13
3.2	Composer-osajärjestelmä	13
3.2.1	StaticTTK91Composer	13
3.2.2	DynamicTTK91Composer	15
3.2.3	FillInTTK91Composer	15

3.2.4	TTK91TaskParser	15
3.2.5	TTK91SyntaxChecker	16
3.2.6	TaskDefinitionController	19
3.3	Analysier-osajärjestelmä	19
3.3.1	Answer2	21
3.3.2	CommonAnalysier implements AnalysierInterface	21
3.3.3	StaticTTK91Analysier extends CommonAnalysier	22
3.3.4	FillInTTK91Analysier extends StaticTTK91Analysier	23
3.3.5	DynamicTTK91Analysier extends StaticTTK91Analysier	23
3.3.6	TTK91RealAnalysier	23
3.3.7	TTK91AnalysierExtraUtils	24
3.3.8	TTK91AnalyseData	25
3.3.9	TTK91FeedbackComposer	27
3.3.10	TTK91AnalyseResults	27
3.3.11	TTK91AnalysierUtils	29
3.3.12	Control (Titokoneen rajapintaluokka)	29
3.4	Osajärjestelmille yhteiset luokat	29
3.4.1	TTK91Constant	29
3.4.2	TTK91TaskCriteria	31
3.4.3	InvalidTTK91CriteriaException	32
3.4.4	TTK91TaskOptions	32
3.4.5	TTK91TaskFeedback	34
3.4.6	TaskBase implements AttributeCache	37
4	Käyttöliittymä	37
4.1	Composer – tehtävän luonti ja muokkaus	37

	iii
4.1.1 Kriteerien syntaksi	40
4.1.2 Symbolien merkitykset kriteerimäärittelyssä	40
4.1.3 Dynaamisen tehtävän määrittelyn syntaksi	40
4.2 Displayer – tehtävän ratkaisu ja palaute	41
4.3 Analyser – vastauksen analysointi	41
5 Muutokset ja lisäykset ulkoisiin osajärjestelmiin	42
5.1 CommonAnalyserin näkyvyysmäärittely muutettu	42
5.2 Lisäykset eAssariin	42
5.3 Lisäykset Titokoneeseen	51
Lähteet	51

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.

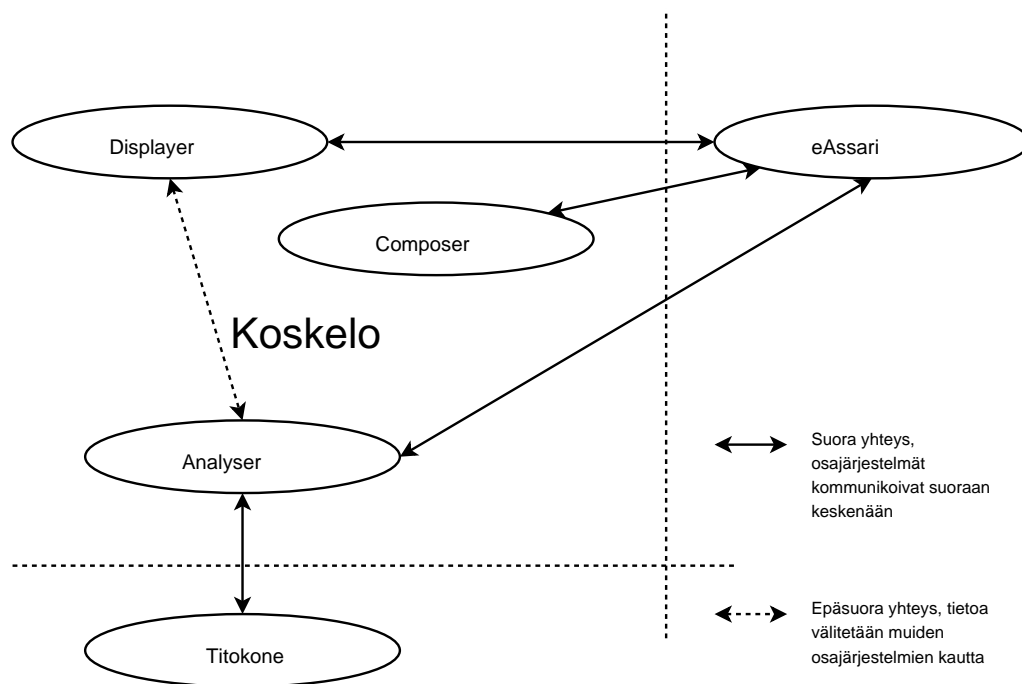
Tietokoneen toiminta -kurssilla käsitellään mm. kuvitteellisen TTK-91-koneen [Häk91] symbolista konekieltä, jonka simulaattori Titokone on. eAssari tarjoaa työkalut yksinkertaisten harjoitustehtävien automaattiseen tarkastamiseen. Koskelo-projektissa yhdistetään eAssariin Titokoneen toiminnallisuutta,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

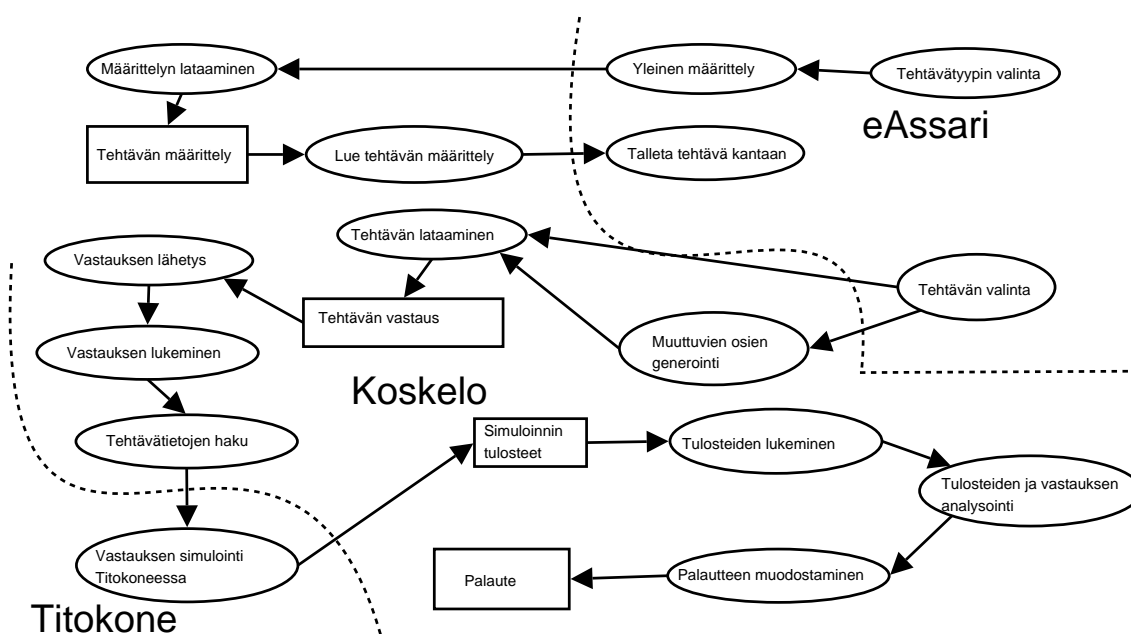
eAssari Automaattisesti tarkastettavat harjoitustehtävät mahdollistava järjestelmä, jota Koskelo-projekti laajentaa TTK-91-ohjelmointitehtäytyypeillä.

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

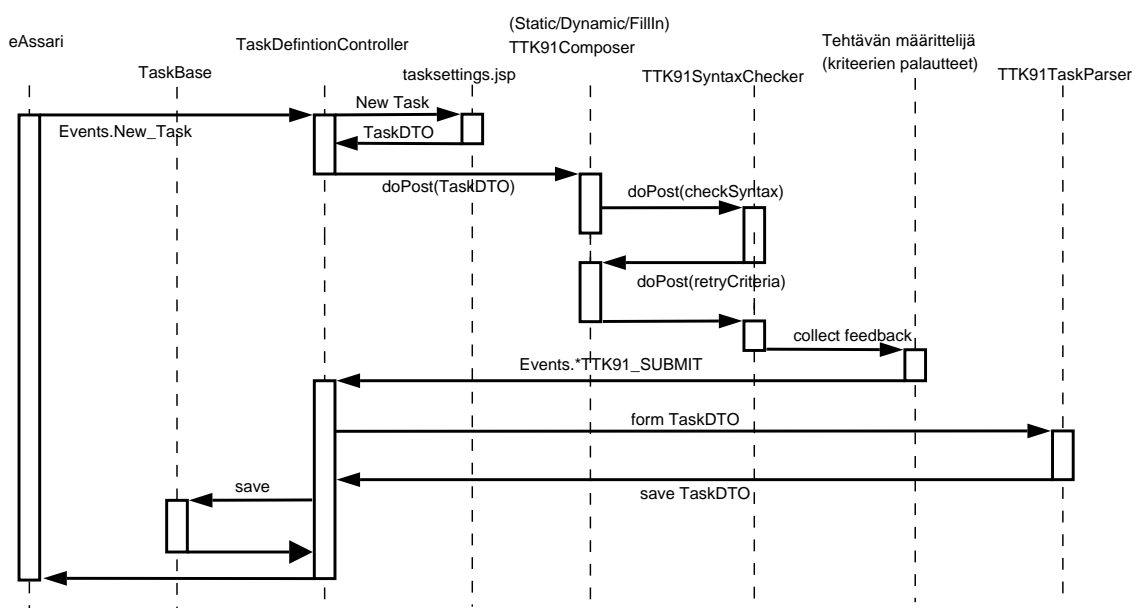
2.1 Sisäiset osajärjestelmät

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. Tehtävämäärittelyä Composerissa esittää sekvenssikaavio kuva 3.

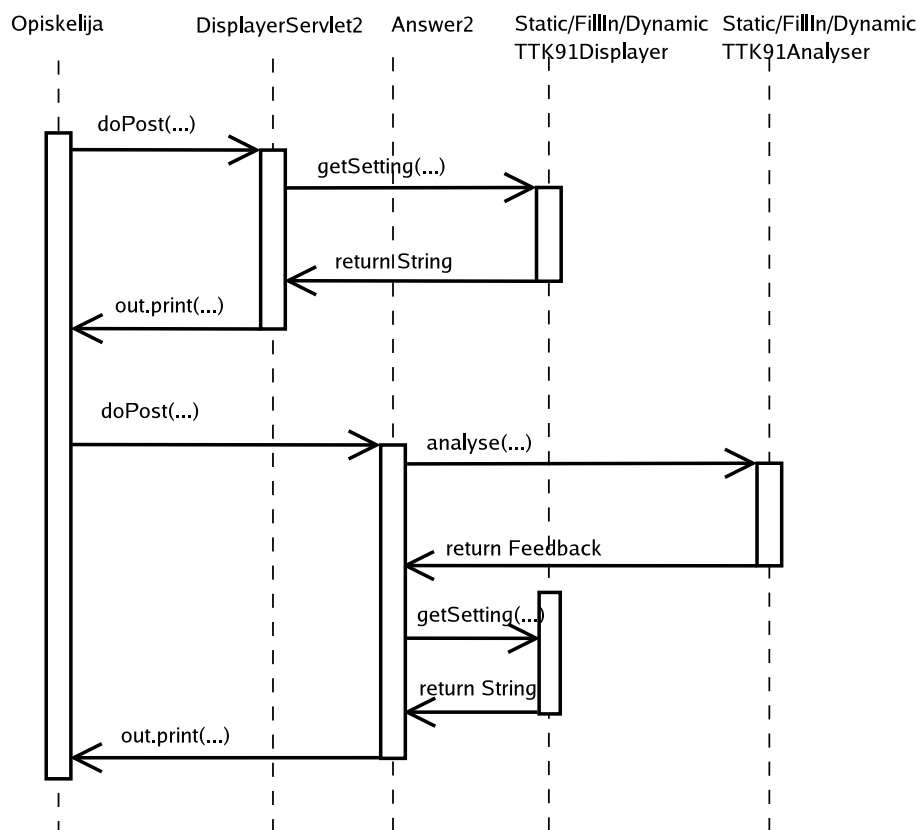


Kuva 3: Tehtävämäärittelyn sekvenssikaavio

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, tehtävänlaati-ajan 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.



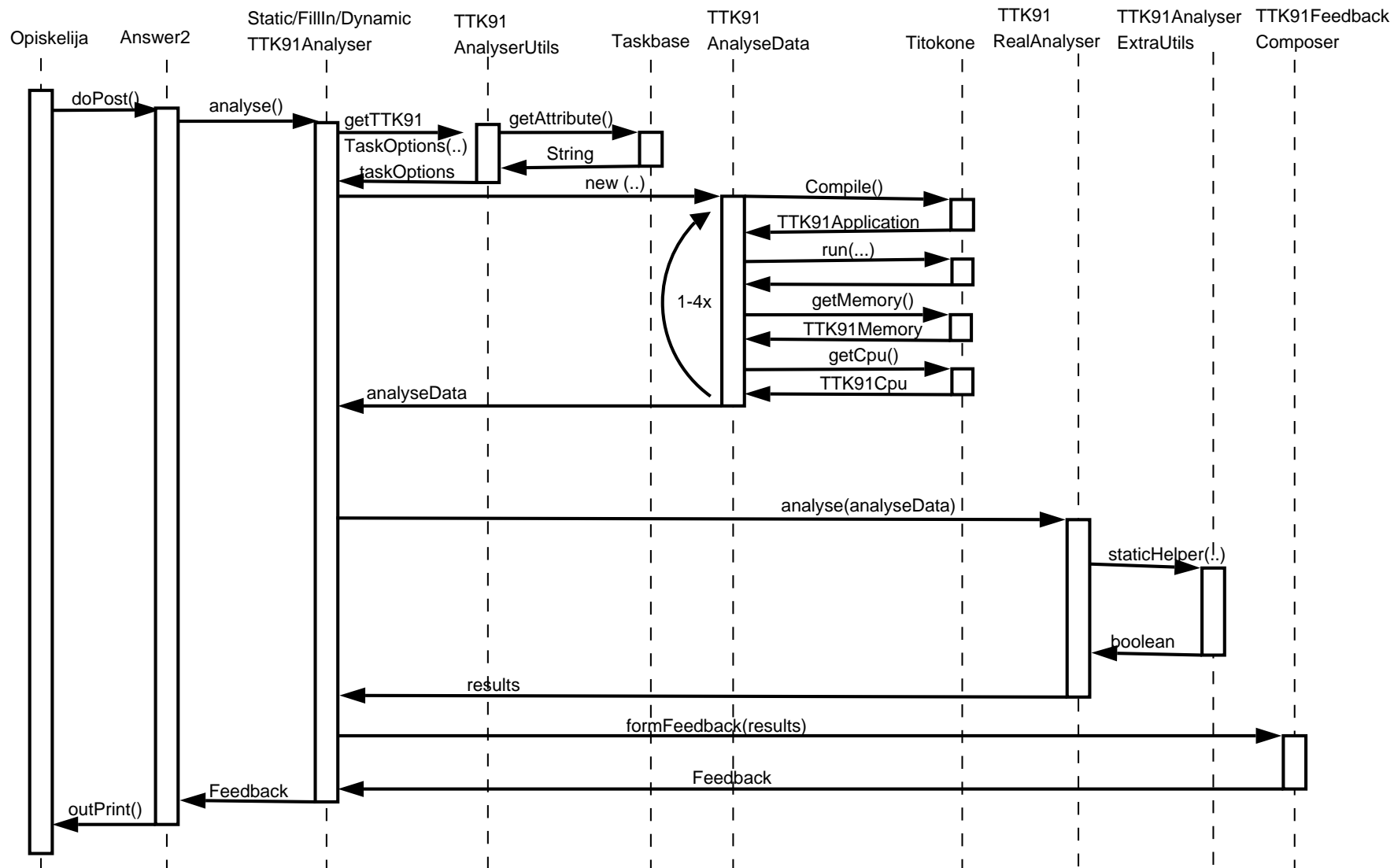
Kuva 4: Tehtävän esityksen ja tarkastuksen sekvenssikaavio

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

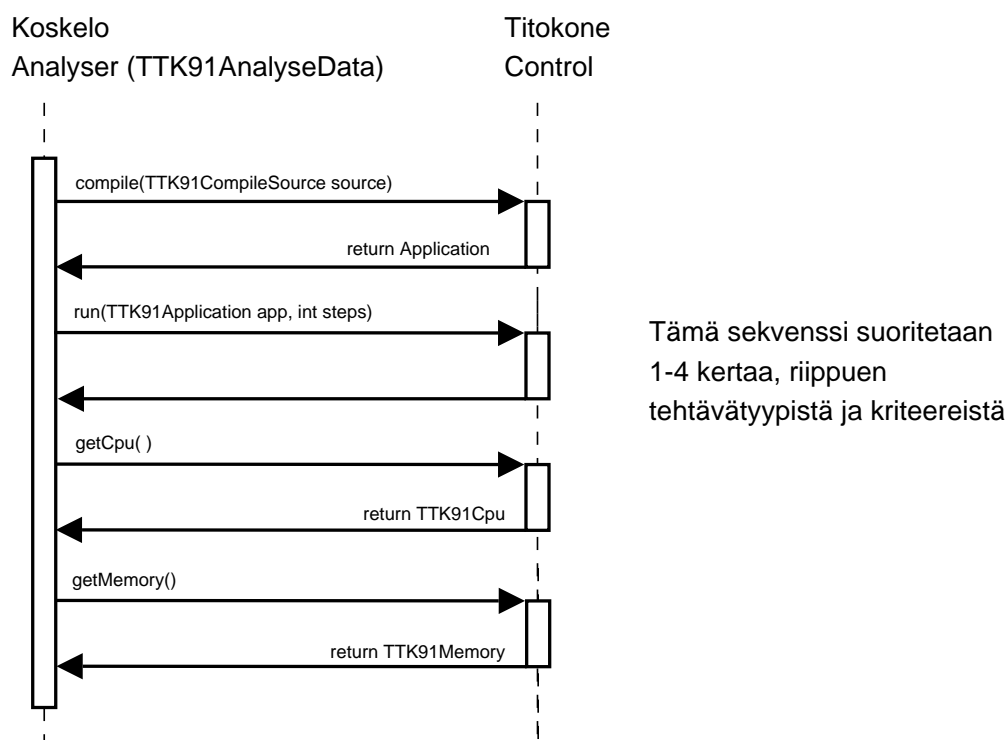


2.2 Ulkoiset osajärjestelmät

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.



Kuva 6: Analyserien ja Titokoneen yhteistyön sekvenssi

2.2.2 eAssari

Ulkoisen 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 tehtävän 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.

3.1.1 CommonDisplayer implements DisplayerInterface

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

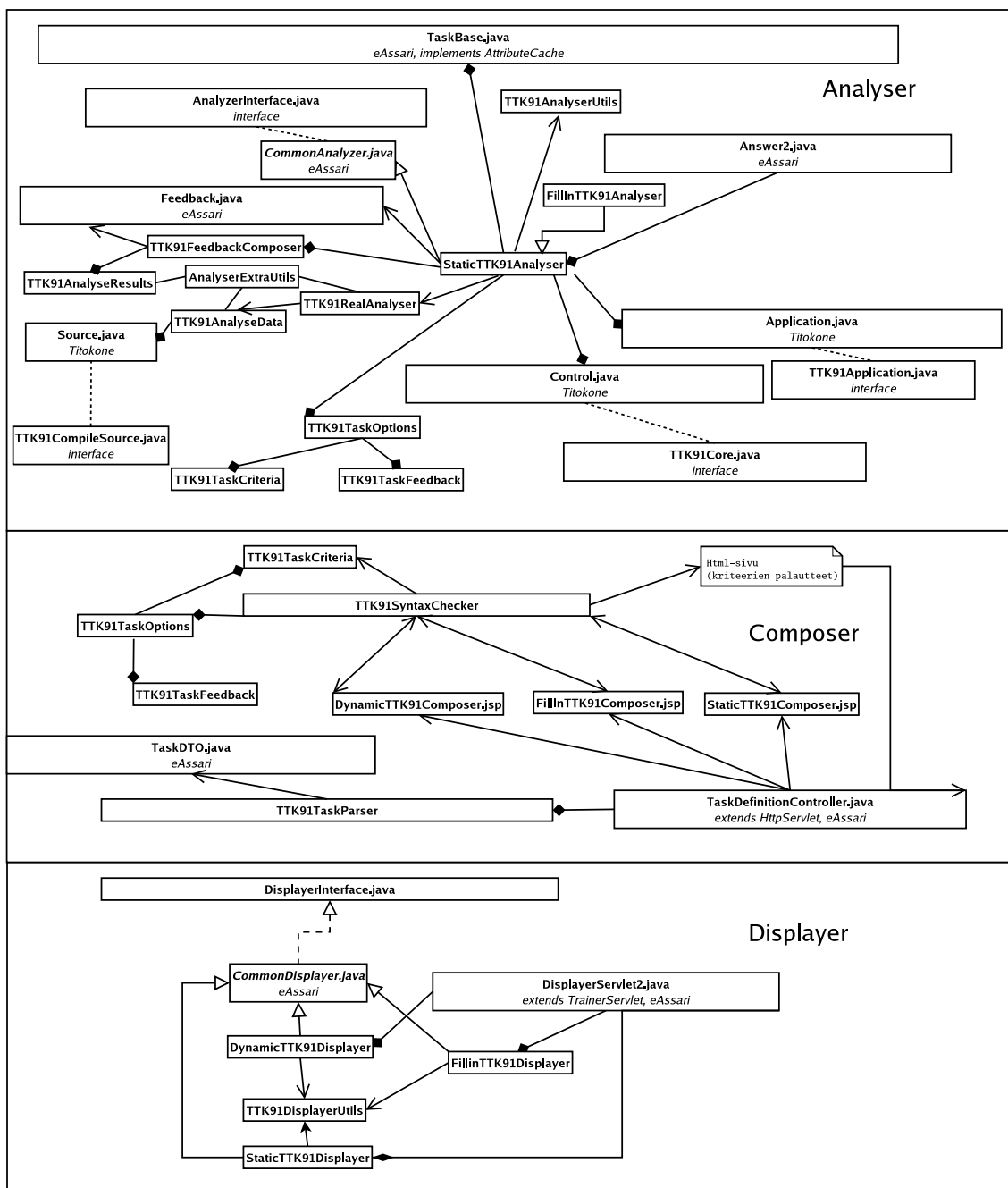
```
public String init(String taskid, String language, String initparams)
```

```
public String getScript()
```

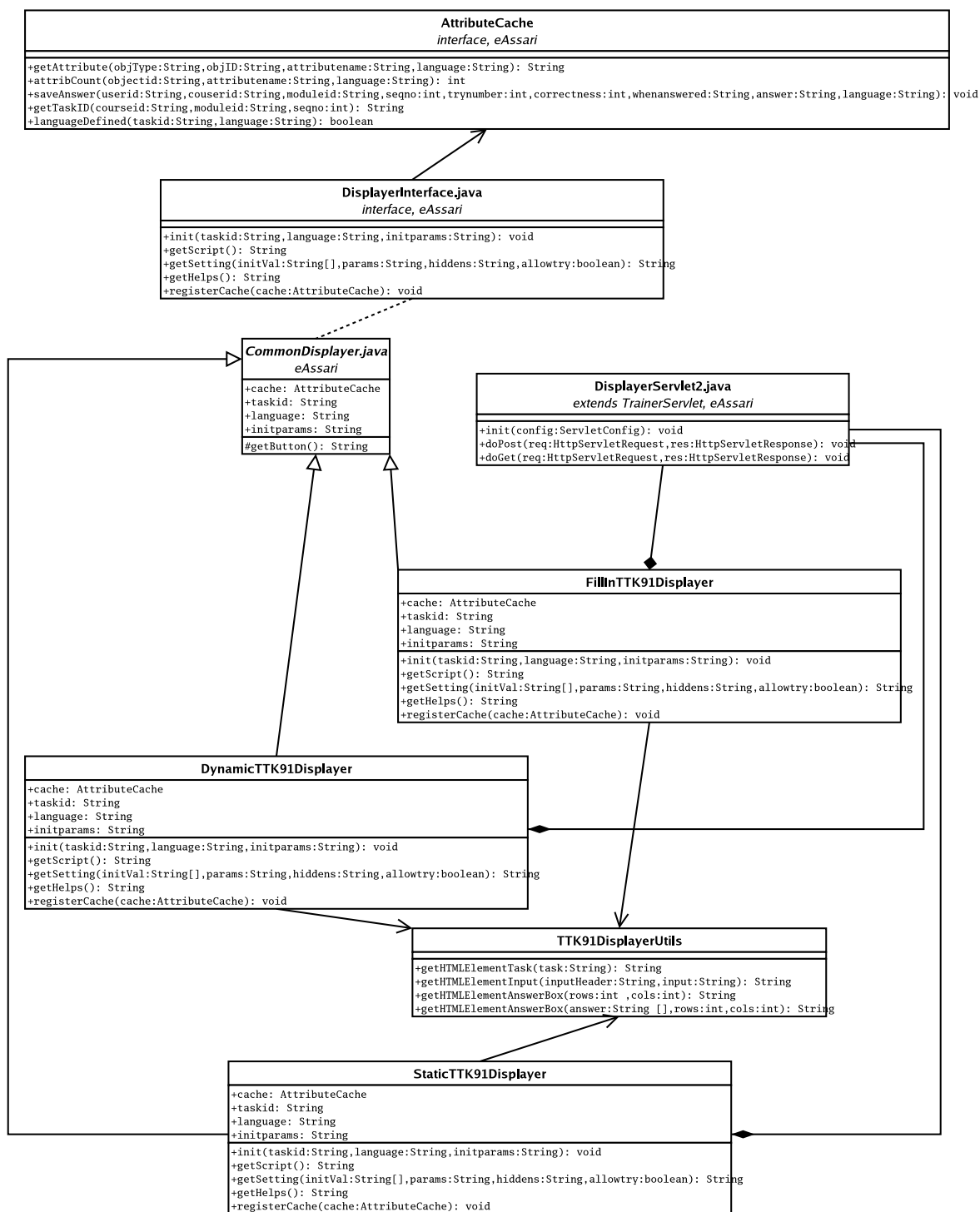
```
public void getSetting(String initVal[], String params, String hidden, boolean allowtry)
```

```
public String getHelps()
```

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



Kuva 7: Luokkarakenne osajärjestelmittäin



Kuva 8: Displayer osajärjestelmä

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()
```

eAssarin rajapinnan vaatima metodi, jota edes eAssari ei käytä. Ei toteuteta.

```
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()
```

eAssarin rajapinnan vaatima metodi, jota edes eAssari ei käytä. Ei toteuteta.


```
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 tunnisteen sekä alkuarvot.

```
public String getScript()
```

Palauttaa sivulla käytettävän java-script koodin, jos sellaista on.

```
public void getSetting(String initVal[], String params, String hid-
```

dens, boolean allowtry)

Palauttaa Displayerin koostaman HTML-sivun String-oliona.

```
public String getHelps()
```

eAssarin rajapinnan vaatima metodi, jota edes eAssari ei käytä. Ei toteuteta.

```
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 , joita on helppo liittää toisiinsa kun muodostetaan sivua.

```
public String getHTMLElementTask(String task)
```

Luo parametrina annetusta tehtävästä tehtävänannon HTML-muotoisena.

```
public String getHTMLInputElement(String inputHeader, String input)
```

Saa parametreina syötteen otsikon ja syötteen, joiden perusteella metodi luo syötekentän HTML-muodossa.

```
public String getHTMLElementAnswerBox(int rows, int cols)
```

Luo tyhjän HTML-muotoisen vastauskentän.

```
public String getHTMLElementAnswerBox(String [] answer, int rows, int
```

cols)

Luo HTML-muotoisen vastauskentän, joka sisältää parametrina annetun tehtävän.

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 parametreilla 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.

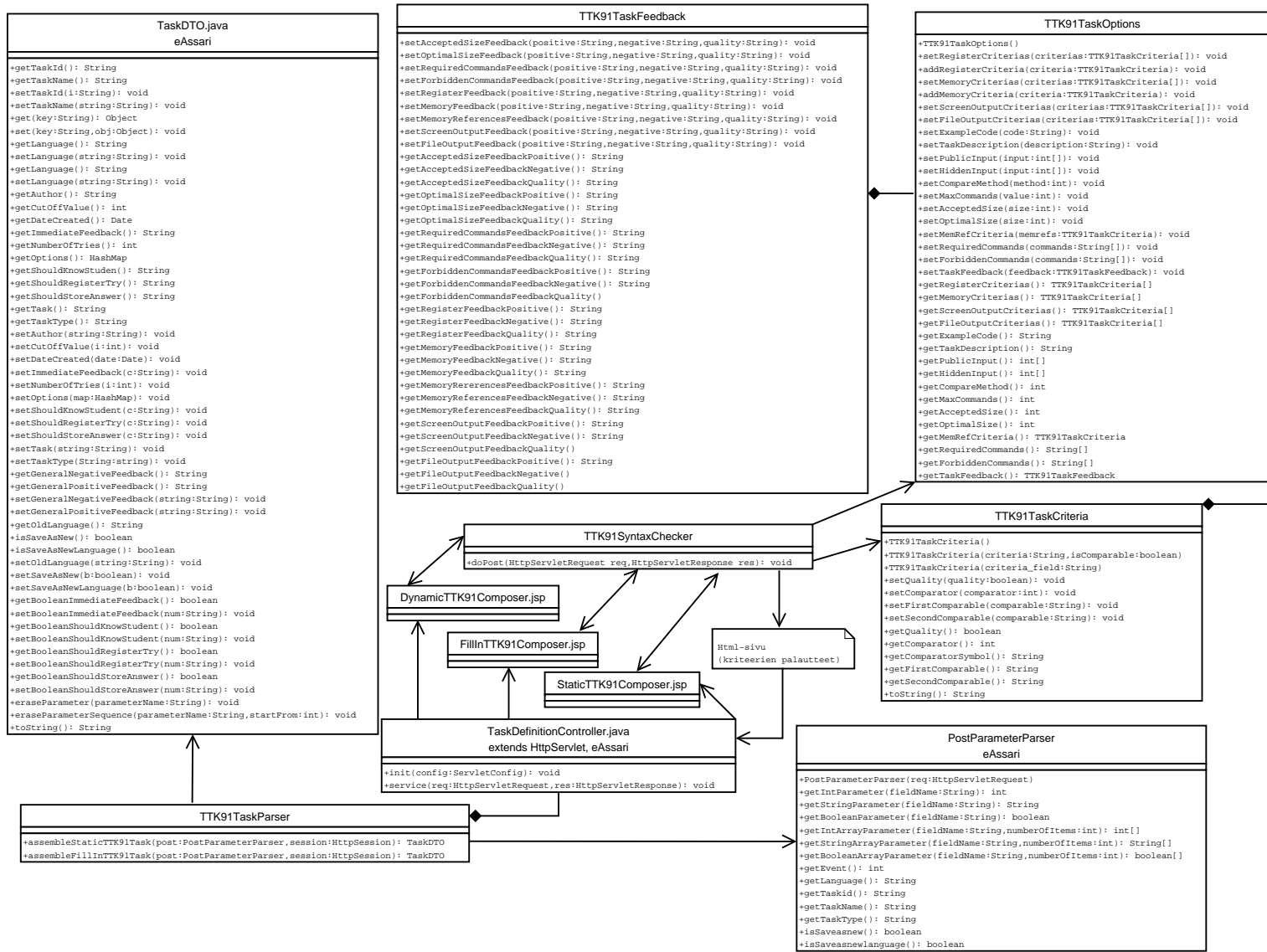
3.2.1 StaticTTK91Composer

JSP-sivu joka toteuttaa staattisen TTK91-tehtävämäärittelyn käyttöliittymän. Sivulla voi sekä luoda uuden tehtävän, että muuttaa vanhaa. Sivun kutsuu syntaksin tarkistamiseksi servlettia 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 julkiset kentät jotka lähetetään SyntaxCheckerille:

- Malliratkaisu: exampleCode
- Tehtävänanto: taskDescription
- Opiskelijan syötteet: publicInput
- Salaiset syötteet: hiddenInput
- Vertailuperuste (malliratk. vai kiinteät arvot): compareMethod

Kuva 9: Composer-osajärjestelmän luokkakaavio



- 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`

Lähetettävän lomakkeen salaiset kentät on kuvattu Assari-projektin [Ar04] suunnitteludokumentissa. Tehtävän editointi erotetaan uuden tehtävän luonnista tutkimalla onko tarjolla `TaskDTO`-oliota sessiossa. Tämä noudattaa Assari-projektin mallia.

3.2.2 `DynamicTTK91Composer`

Kuten staattisen tehtävän määrittelevä jsp-sivu, mutta tulostaa sivun, jossa on laajennuksia dynaamisten tehtävien tietojen täyttämiseksi. Dynaamista tehtävätyyppiä ei projektin puitteissa ehditty toteuttaa.

3.2.3 `FillInTTK91Composer`

Kuten staattisen tehtävänmäärittelyn jsp-sivu.

3.2.4 `TTK91TaskParser`

Apuluokka `TaskDefinitionController`ia varten. Koostaa annetusta datasta `TaskDTO`-olion. `TTK91TaskParser` saa parametrinä `PostParameterParser`in sekä `HttpSession`. `PostParameterParser` tarjoaa post-kutsun mukana tulleet palaute-kentät ja `HttpSession`ista saadaan `TTK91TaskOptions`-olio joka sisältää kaiken muun tehtävän määrittelyn.

```
public static TaskDTO assembleStaticTTK91Task(PostParameterParser
post, HttpSession session)
```

Kokoaa TTK91-ohjelmointitehtävän.

```
public static TaskDTO assembleFillInTTK91Task(PostParameterParser
post, HttpSession session)
```

Kokoaa TTK91-täydennystehtävän. Käytännössä sama toiminta kuin assembleStaticTTK91Taskilla.

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, palauttaa TTK91SyntaxChecker virheen. Kriteerit palautetaan takaisin TTK91Composerille, joka ilmoittaa tehtävänlaatijalle jonkin kriteerin olevan virheellinen. Composer erottaa virhetilanteen uuden tehtävän luonnista sessio-oliossa olevan kentän TTK91ERROR avulla. Virhetilanteessa sen arvona on viesti miksi tehtävää ei kelpuutettu. Muussa tapauksessa TTK91ERROR on null.

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:	Palaute annetaan kun:
registerFeedbackPositive	Kaikki kriteerit, jotka määrittelevät opiskelijan ratkaisuun liittyviä rekisterien arvoja on täytetty.
memoryFeedbackPositive	Kaikki kriteerit, jotka määrittelevät yksittäisten muistipaikkojen sisältöön liittyviä arvoja on täytetty.
screenOutputFeedbackPositive	Opiskelijan ohjelman näyttötu- losteeet vastaavat vaadittuja.
fileOutputFeedbackPositive	Opiskelijan ohjelman tiedostotu- losteeisiin vastaavat vaadittuja.
requiredCommandsFeedbackPositive	Opiskelijan ohjelmassa esiintyvät kaikki vaaditut konekäskyt.
forbiddenCommandFeedbackPositive	Opiskelijan ohjelmassa ei esiinny yhtään kiellettyä konekäskyä.
acceptedSizeFeedbackPositive	Opiskelijan ohjelman rivimäärä alittaa hyväksytyn koon.
optimalSizeFeedbackPositive	Opiskelijan ohjelma alittaa opti- mikoon.
memoryReferencesFeedbackPositive	Opiskelijan ohjelma täyttää muis- tiviitteisiin liittyvän vaatimuk- sen.

Positiivinen palaute annetaan, jos opiskelijan ratkaisu täyttää kaikki kyseiseen kriteeriryhmään kuuluvat kriteerit. Kuitenkin, jos ryhmään ei ole määritelty yhtään kriteeriä, palautetta ei näytetä. Yksi kenttä sisältää yhden kriteeriryhmän palautteen.

Negatiiviset:	Näytetään kun:
registerFeedbackNegative	Jokin määritellyistä rekisterien arvoista ei täsmää opiskelijan ratkaisussa.
memoryFeedbackNegative	Jokin määritellyistä muistipaikkojen arvoista ei täsmää opiskelijan ratkaisussa.
screenOutputFeedbackNegative	Opiskelijan ohjelma tulosti näytölle jotain muuta kuin tehtävänmäärittelyssä vaadittiin.
fileOutputFeedbackNegative	Opiskelijan ohjelma tulosti tiedostoon jotain muuta kuin tehtävänmäärittelyssä vaadittiin.
requiredCommandsFeedbackNegative	Opiskelijan ohjelma ei käyttänyt vähintään yhtä vaadituista käskyistä.
forbiddenCommandsFeedbackNegative	Opiskelijan ohjelma käytti vähintään yhtä kiellettyä käskyä.
acceptedSizeFeedbackNegative	Opiskelijan ohjelmassa oli enemmän rivejä kuin hyväksymisraja sallii.
optimalSizeFeedbackNegative	Opiskelijan ohjelma ei ole aivan optimikokoinen.
memoryReferencesFeedbackNegative	Opiskelijan ohjelma käyttänyt liian monta muistiviittausta.

Negatiivinen palaute annetaan kun opiskelijan ohjelma ei täytä vähintään yhtä kunkin kriteeriryhmän kriteeriä, jolloin ratkaisu hylätään.

Laadulliset:	Näytetään kun:
registerQualityFeedback	Kaikki rekistereihin liittyvät laadulliset kriteerit täyttyvät.
memoryQualityFeedback	Kaikki muistiin liittyvät laadulliset kriteerit täyttyvät.
screenOutputQualityFeedback	Kaikki laadulliseksi merkityt tulosteet esiintyvät oikeilla paikoillaan.
fileOutputQualityFeedback	Kaikki laadulliseksi merkityt tulosteet esiintyvät oikeilla paikoillaan.
requiredCommandsQualityFeedback	Kaikki laadullisiksi merkityt konekäskyt löytyvät opiskelijan ratkaisusta.
forbiddenCommandsQualityFeedback	Mitään laadulliseksi merkittyä konekäskyä ei esiinny opiskelijan ratkaisussa.

Laadullinen palaute näytetään, kun kussakin kriteeriryhmässä annetut laadulliset kriteerit täyttyvät. Laadullista palautetta ei anneta, jos ryhmässä ei ole määritelty yhtään laadullista kriteeriä. Laadullinen palaute näytetään vain silloin kun kaikki oikeellisuuskriteerit täyttyvät.

3.2.6 TaskDefinitionController

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

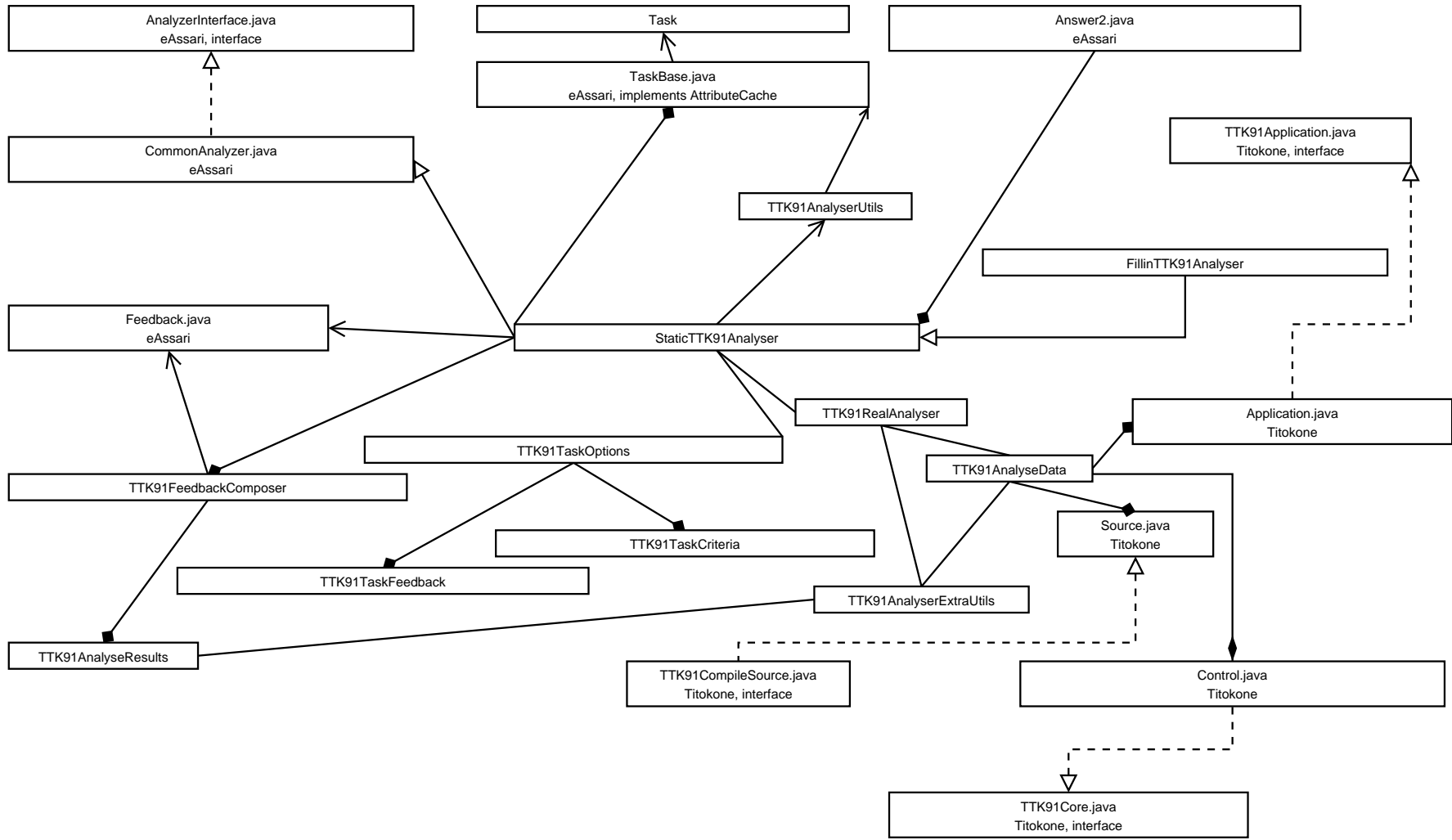
Servletti saa parametrina kokonaisluvun, joka kuvaa tapahtumaa. Tapahtumasta riippuen servletti käynnistää muita palveluita tai kutsuu ulkoisia osajärjestelmiä. Kaikki tehtävänmäärittelyyn liittyvät tapahtumat kulkevat tämän servletin kautta.

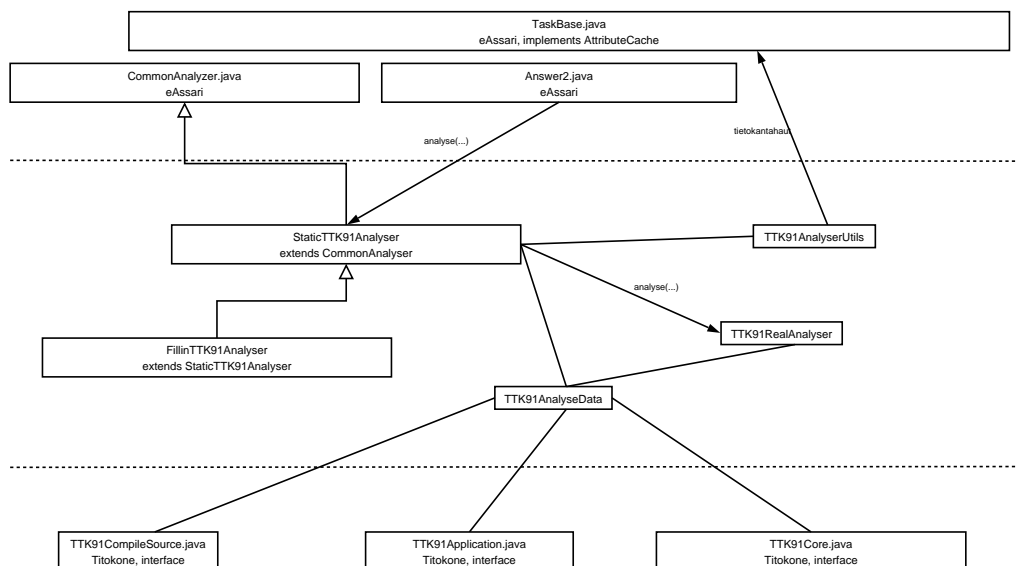
Myös TaskDefinitionController on servletti ja sen kautta eAssari käynnistää oikean tehtävänmäärittelysivun. Tehtävänmäärittelysivut (eri Composerit) on toteutettu jsp-sivuina.

3.3 Analyser-osajärjestelmä

Analyser-osajärjestelmä toimii yhteistyössä ulkoisen osajärjestelmän, Titokoneen kanssa. Analyser tutkii opiskelijan ratkaisuyritysten oikeellisuutta, ja antaa niihin liittyvän palautteen. Titokonetta Analyser käyttää ratkaisuna annettujen TTK91-ohjelmien simulointiin. Analyserin rakennetta kuvaava luokkakaavio kuvassa 10.

Kuva 10: Analyser-osajärjestelmän luokkakaavio





Kuva 11: Analyser-osajärjestelmän rajapinnat ja sisäinen periytymishierarkia

Analyser-osajärjestelmä on yhteydessä Titokoneen lisäksi myös toiseen ulkoiseen osajärjestelmään, eAssariin. Kuvassa 11 Analyserin rajapinnat ulkoisiin osajärjestelmiin ja sisäinen periytymishierarkia.

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 12.

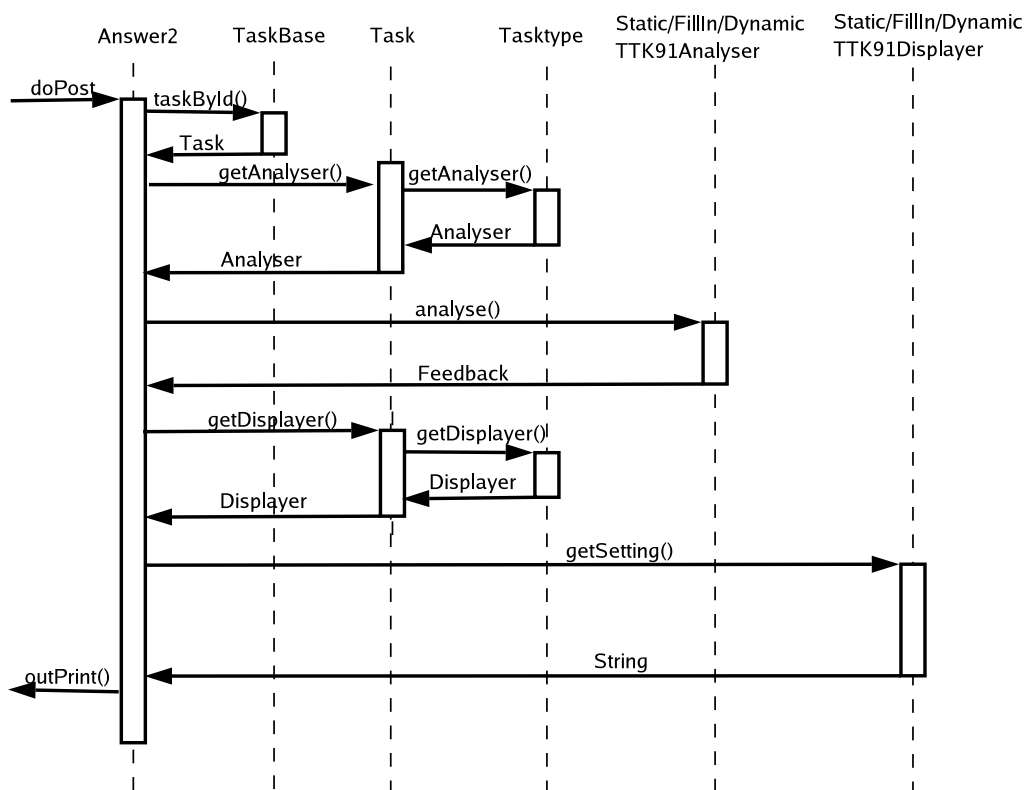
3.3.2 CommonAnalyser implements AnalyserInterface

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 initparams)
```

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

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



Kuva 12: Answer2-luokan toiminta

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 (salaisia) syötteitä. TTK91-ohjelman suoritus tapahtuu TTK91AnalyseData-luokassa, Control-luokan kautta. Titokonesimulaation jälkeen StaticTTK91Analyser antaa TTK91AnalyseData-ilmentymän TTK91RealAnalyserille varsinaista analysointia varten. TTK91RealAnalyserilta saadaan TTK91AnalyseResult-luokan ilmentymä, jonka StaticTTK91Analyser antaa TTK91FeedbackComposerille. TTK91AnalyseResultsista TTK91FeedbackComposer muodostaa Feedback-olion.

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

```
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 viiteen AttributeCache-rajapinnan toteuttavaan TaskBase-luokkaan. Tämän jälkeen voidaan hakea tehtävään kuuluvia attribuutteja getAttribute-metodilla. Samalla noudetaan TTK91AnalyserUtils-luokan ilmentymä ja TTK91TaskOptions-ilmentymä. Nämä noudot tehdään, jotta aliluokat (kuten FillinTTK91Analyser) saavat käyttövalmiin ympäristön (esimerkiksi tietokantayhteys TTK91AnalyserUtils-ilmentymän avulla)

```
protected String[] fetchExampleCode()
```

Palauttaa mahdollisen malliratkaisun. Peitetään esimerkiksi FillinTTK91Analyserissa, jolloin voidaan muokata esimerkkikoodi toimivaksi poistamalla opiskelijalle näkymättömät osat.

3.3.4 FillInTTK91Analyser extends StaticTTK91Analyser

Toimii muuten kuten StaticTTK91Analyser, mutta arvioi täydennystehtävän ratkaisun. Sisältää samat metodit kuin StaticTTK91Analyser. Peittää fetchExampleCode-metodin omalla toteutuksellaan

3.3.5 DynamicTTK91Analyser extends StaticTTK91Analyser

Toimii muuten kuten StaticTTK91Analyser, mutta arvioi dynaamisen tehtävän ratkaisun. Sisältää samat metodit kuin StaticTTK91Analyser. Ei toteutettu Koskelo-projektissa.

3.3.6 TTK91RealAnalyser

Luokka vastauksen analysoinnin pyörittämiseen. Kirjastoluokka TTK91AnalyserExtraUtils tarjoaa apumetodeja vastauksen analysointiin.

```
public TTK91AnalyseResults analyse()
```

Luokan ainoa julkinen metodi. Kutsuu omia yksityisiä apumetodejaan, jotka suorittavat analysoinnin kriteerityypeittäin. Apumetodit hakevat tarvitsemansa tiedot TTK91AnalyseData-luokan ilmentymästä. Apumetodit käyttävät puolestaan kirjastoluokka TTK91AnalyserExtraUtilsin palveluja.

3.3.7 TTK91AnalyserExtraUtils

Kirjastoluokka, josta ei luoda ilmentymiä. Sisältää itsenäisiä staattisia metodeja kriteerityyppien tutkimiseen.

Luokan sisältämät julkiset metodit:

```
public static boolean checkStaticMemoryCriteria(...)
public static boolean checkStaticMemoryCriteria(...)
public static boolean checkSimulatedMemoryCriteria(...)
public static boolean checkSimulatedMemoryCriteria(...)
```

Neljä metodia muistikriteerien tarkastamiseen. Eri tapauksille (verrataanko mallivastaukseen vai suoraan kriteeri-oliossa määriteltyihin arvoihin, käytetäänkö salaisia syötteitä vai ei) omat metodinsa. Palauttavat arvon true, jos ratkaisu täyttää kriteerien vaatimukset, false muuten.

```
public static boolean checkRegisterCriteria(...)
```

Tarkastaa rekisterien arvoihin liittyvät kriteerit. Palauttaa true, jos ratkaisu täyttää kriteerien vaatimukset, false muuten.

```
public static boolean checkOutputCriteria(...) public static boolean
checkOutputCriteria(...)
```

Kaksi metodia näyttötulosteiden (CRT) ja tiedostotulosteiden tarkastamiseen.

```
public static boolean compare(int first, int comparator, String second)
public static boolean compare(int first, int comparator, int second)
```

Vertailuoperaatiot. String-parametrin ottava metodi osaa itsenäisesti muuntaa merkkijonoparametrin lukuarvoksi.

```
public static Boolean isCommandFound(String answer, String cmd)
```

Vastaa kysymykseen *Sisältääkö answer merkkijonoa cmd?*.

```
public static Boolean commandNotFound(String answer, String cmd)
```

Käänteinen edelliseen verrattuna.

```
public static int[] parseOutputArrays(String from)
```

Titokoneen rajapinnan kautta tulosteet (numeroarvoja) saadaan merkkijonoina. Tämä metodi muuntaa merkkijonot int-taulukoiksi.

3.3.8 TTK91AnalyseData

Kerää tehtävän analysoinnissa tarvittavat tiedot, ja jakaa ne varsinaiselle analysointikoneistolle. Luokan ilmentymää luotaessa suoritetaan tarvittavat simuloinnit Titokoneella.

Konstruktori `public TTK91AnalyseData(TTK91TaskOptions to, String answer, String exampleCode)`

Luo uuden TTK91AnalyseData-ilmentymän. Konstruktori suorittaa varsinaiset simuloinnit Titokoneella.

TTK91AnalyseData:n tiedot saa julkisilla get-metodeilla:

`public String getStudentCompileError()`

Palauttaa mahdolliset Titokoneen käännösaikaiset virheilmoitukset käännettäessä opiskelijan ratkaisua.

`public String getTeacherCompileError()`

Palauttaa mahdolliset Titokoneen käännösaikaiset virheilmoitukset käännettäessä malliratkaisua.

`public String getStudentRunError()`

Palauttaa mahdolliset Titokoneen ajonaikaiset virheilmoitukset opiskelijan ratkaisua simuloitaessa.

`public String getTeacherRunError()`

Palauttaa mahdolliset Titokoneen ajonaikaiset virheilmoitukset malliratkaisua simuloitaessa.

`public String[] getErrorMessages()`

Palauttaa kaikki edellämainitut virheilmoitukset String-taulukkona.

`public boolean errors()`

Palauttaa true, jos käännö- tai ajonaikaisia virheitä.

`public boolean compareToHidden()`

Käytetäänkö vaihtoehtoisia, opiskelijalle näkymättömiä syötteitä simulaatiossa.

`public String getAnswer()`

Palauttaa opiskelijan vastauksen.

`public TTK91Application getStudentAppPublic()`

`public TTK91Application getTeacherAppPublic()`

`public TTK91Application getStudentAppHidden()`

`public TTK91Application getTeacherAppHidden()`

Neljä metodia, jotka palauttavat binäärimuotoisen TTK91-sovelluksen. Eri vaihtoehdot eri tilanteille, eli ollaanko vertaamassa malliratkaisuun vai suoraan staattisesti

ennaltamäärättyihin kriteereihin, ja onko käytössä piilosyötteitä.

```
public TTK91Cpu getStudentCpuPublic()
public TTK91Cpu getStudentCpuHidden()
public TTK91Cpu getTeacherCpuPublic()
public TTK91Cpu getTeacherCpuHidden()
```

Neljä metodia, jotka palauttavat simuloinnissa käytetyn TTK91Cpu-suorittimen. Niistä saadaan mm. rekisterien arvot suorituksen jälkeen. Eri vaihtoehdot eri tilanteille, eli ollaanko vertaamassa malliratkaisuun vai suoraan staattisesti ennaltamäärättyihin kriteereihin, ja onko käytössä piilosyötteitä.

```
public TTK91Memory getStudentMemoryPublic()
public TTK91Memory getStudentMemoryHidden()
public TTK91Memory getTeacherMemoryPublic()
public TTK91Memory getTeacherMemoryHidden()
```

Neljä metodia, jotka palauttavat TTK91Memory-olion, jotka kuvaavat Titokoneen muistia. Eri vaihtoehdot eri tilanteille, eli ollaanko vertaamassa malliratkaisuun vai suoraan staattisesti ennaltamäärättyihin kriteereihin, ja onko käytössä piilosyötteitä

```
public TTK91TaskOptions getTaskOptions()
```

Palauttaa tähän tehtävään liittyvän TTK91TaskOptions-ilmentymän. Siitä saadaan mm. ratkaisun analysoinnissa käytettävät kriteerit.

```
public int getCommandAmount()
```

Palautusarvona julkisilla syötteillä suoritettujen opiskelijan ratkaisun konekäskyjen lukumäärä.

```
public int getHiddenCommandAmount()
```

Palautusarvona piilosyötteillä suoritettujen ohjelman konekäskyjen lukumäärä.

```
public int getStackSize()
```

Palauttaa pinon maksimikoon opiskelijan ratkaisun (julkisilla syötteillä) simuloinnissa.

```
public int getMemoryReferences()
```

Palauttaa muistiviitteiden määrän opiskelijan ratkaisun (julkisilla syötteillä) simuloinnissa.

```
public int getCodeSegmentSize()
```

Palauttaa koodisegmentin koon opiskelijan ratkaisun (julkisilla syötteillä) simuloinnissa.

```
public int getDataSegmentSize()
```

Palauttaa datasegmentin koon opiskelijan ratkaisun (julkisilla syötteillä) simuloinnissa.

```
public int analyseSwitcher()
```

Palautusarvo kertoo miten simuloinnit on ajettu. Onko piilosyötteitä ja simuloitiinko malliratkaisu.

```
public String getStudentScreenOutputPublic()
public String getStudentScreenOutputHidden()
public String getTeacherScreenOutputPublic()
public String getTeacherScreenOutputHidden()
public String getStudentFileOutputPublic()
public String getStudentFileOutputHidden()
public String getTeacherFileOutputPublic()
public String getTeacherFileOutputHidden()
```

Palauttavat opiskelijan ratkaisun (student) ja malliratkaisun (teacher) simuloinnissa syntyneet tulosteet. Eri metodit näyttötulosteille (Screen eli CRT) ja tiedostotulosteille (File).

3.3.9 TTK91FeedbackComposer

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

```
public static Feedback formFeedback(TTK91AnalyseResults analyseResults,
TTK91TaskFeedback taskFeedback, AttributeCache cache, String taskID,
String language) throws CacheException
```

Parametri analyseResults sisältää tehtävän analysoinnin tulokset. Parametri taskFeedback sisältää kaiken tehtävään liittyvän palautteen. Cachesta haetaan käyttöliittymä komponentteja. TaskID on käsiteltävän tehtävän tunnus ja language sen kieli.

Palaute muodostuu cachesta haetuista kielikohtaisista käyttöliittymäelementeistä ja TTK91TaskFeedback-luokasta haetuista kriteerien palautteista.

```
public static Feedback formFeedback(String errorMessage)
```

Metodi palauttaa Feedback-olion, joka sisältää virheilmoituksen.

3.3.10 TTK91AnalyseResults

Luokka, johon voidaan analysoinnin edetessä vaiheittain koota kustakin kriteeriryhmästä niiden toteutuminen. Set-metodien boolean arvot ovat: true jos kriteeri täyttyy, false jos kriteeri ei täyty ja null jos kriteeriä ei ole määritetty. Lisäksi luokkaan kootaan Titokoneen suorituksen statistiikka. Jokaista set-metodia kohden löytyy vastaava get-metodi. Luokka sisältää seuraavat set-metodit:

```
public void setRegisters(boolean b)
```



```
public void setMemory(boolean b)
public void setScreenOutput(boolean b)
public void setFileOutput(boolean b)
public void setRequiredCommands(boolean b)
public void setForbiddenCommands(boolean b)
public void setAcceptedSize(boolean b)
public void setMemoryReferences(boolean b)
public void setRegistersQuality(boolean b)
public void setMemoryQuality(boolean b)
public void setScreenOutputQuality(boolean b)
public void setFileOutputQuality(boolean b)
public void setRequiredCommandsQuality(boolean b)
public void setForbiddenCommandsQuality(boolean b)
public void setOptimalSize(boolean b)
public void setMemoryReferencesQuality(boolean b)
public void setRegisterValues(int[] registers)
public void setMemoryValues(int[] memory)
public void setMemoryReferenceCount(int memoryReferences)
public void setStackSize(int stackSize)
public void setCodeSegmentSize(int codeSegmentSize)
public void setDataSegmentSize(int dataSegmentSize)
public void setExecutedCommandsCount(int executedCommands)
public void setCrt(int[] crt)
public void setFile(int[] file)
```

3.3.11 TTK91AnalyserUtils

Kutsutaan kaikista TTK91Analyser-luokista. Luokka sisältää kaikille tehtävätyypeille yhteisiä metodeja eAssarin cache-tietorakenteen käsittelyyn.

```
public TTK91TaskOptions getTTK91TaskOptions()
```

Hakee kaikki kriteerit cachesta ja asettaa ne TTK91TaskOptions-luokan ilmentymään.

3.3.12 Control (Titokoneen rajapintaluokka)

Kuvassa 6 (sivulla 7) Titokoneen Control-rajapintaluokan ulkoinen toiminta. Tehtäväkohtaiset Analyserit kutsuvat Titokoneen Control-rajapintaluokan metodeja. Ensin pyydetään käännoispalvelua, joka palauttaa ajettavan TTK91-ohjelman. Tämä ohjelma laitetaan suoritukseen, jonka jälkeen suorituksesta pyydetään tiedot (vrt.5.3).

```
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 Processor-luokasta.

```
public TTK91Memory getMemory()
```

Palauttaa ilmentymän Titokoneen RancamAccessMemory-luokasta.

3.4 Osajärjestelmille yhteiset luokat

Tämä luku sisältää pakkauksen fi.helsinki.cs.koskelo.common -luokkien kuvaukset.

3.4.1 TTK91Constant

Luokka, joka sisältää Koskelon omissa luokissaan käyttämät vakiot.

```
public final static int COMPARE_TO_SIMULATED=0;
```

Vertailu tehtävää tarkastaessa suoritetaan vertaamalla opiskelijan ratkaisua malli-ratkaisuun.

```
public final static int COMPARE_TO_STATIC=1;
```

Vertailu tehtävää tarkastaessa suoritetaan vertaamalla staattiseen lopputillan.

```
public final static int MAX_COMMANDS=10000;
```

Oletusarvo maksimimäärälle suoritettavia konekäskyjä.

```
public static final int INVALID=-1;
```

Alustamaton vertailu kriteerissä.

```
public static final int LESS=0;
```

<

```
public static final int LESSEQ=1;
```

<=

```
public static final int GREATER=2;
```

>

```
public static final int GREATEREQ=3;
```

>=

```
public static final int EQUAL=4;
```

=

```
public static final int NOTEQUAL=5;
```

!=

```
public static final int NOTCOMPARABLE=6;
```

’,’ Kriteereitä varten joissa ei ole määritelty vertailuoperaattoria. Näissä vertailuoperaattori korvaataan pilkulla.

```
public static final int FATAL_ERROR=2;
```

eAssarin palautteeseen: ratkaisun analysoinnissa tapahtunut vakava virhe.

```
public static final int ERROR=1; eAssarin palautteeseen: ratkaisun analysoin-
```

nissa tapahtunut virhe.

```
public static final int NO_ERROR
```

Tehtävän analysoinnissa ei tapahtunut virhettä.

3.4.2 TTK91TaskCriteria

Kääreluokka joka sisältää yhden kriteerin.

```
public TTK91TaskCriteria()
Luo tyhjän TTK91TaskCriteriaolion
```

```
public TTK91TaskCriteria(String criteriaField)
Asettaa kriteerin ja tutkii samalla sen syntaksin. Heittää InvalidTTK91Criteria-poikkeuksen jos syntaksi virheellinen.
```

```
public TTK91TaskCriteria(String criteriaField, boolean Comparable)
Asettaa kriteerin ja tutkii samalla sen syntaksin. Boolean-operaattorilla on mahdollista ilmaista, että kriteeri ei sisällä varsinaista vertailuoperaattoria, vaan sen sijaan pilkun. Heittää InvalidTTK91Criteria -poikkeuksen jos syntaksi virheellinen.
```

```
public void setQuality(boolean quality)
Asettaa vertailuoperaattorin
```

```
public void setComparator(int comparator)
throws InvalidTTK91CriteriaException
Asettaa vertailuoperaattorin. Aiheuttaa poikkeuksen, jos vertailuoperaattori on jokin muu kuin luokassa TTK91Constant määritellyt vakiot.
```

```
public void setFirstComparable(String comparable)
throws InvalidTTK91CriteriaException
Asettaa vertailun vasemmanpuoleisen alkion.
```

```
public void setSecondComparable(String comparable)
throws InvalidTTK91CriteriaException
Asettaa vertailun oikeanpuoleisen alkion.
```

```
public int getComparator()
Palauttaa vertailuoperaattorin
```

```
public String getComparatorSymbol()
Palauttaa symbolin joka vastaa loogisen operaation kokonaislukuvakiota.
```

```
public String getFirstComparable()
Palauttaa vasemmanpuoleisen vertailuoperaattorin.
```

```
public String getSecondComparable()
Palauttaa oikeanpuoleisen vertailuoperaattorin.
```

```
public boolean getQuality()
Palauttaa tiedon onko kriteeri laadullinen. True on, false ei.
```

```
public String toString()
Palauttaa kriteerin merkkiesityksenä.
```

3.4.3 InvalidTTK91CriteriaException

Poikkeusluokka kriteerin parsinnassa tapahtuvia virheitä varten.

```
public InvalidTTK91CriteriaException()
Tavallinen tyhjä poikkeus.
```

```
public InvalidTTK91CriteriaException(String msg)
Poikkeuksen virheilmoituksen tekstinä on msg.
```

3.4.4 TTK91TaskOptions

Kääreluokka joka sisältää tehtävän kaikki määrittelyt ja kriteerit.

```
public void addRegisterCriteria(criteria TTK91TaskCriteria)
Lisää annetun kriteerin rekisteri-vectorin viimeiseksi.
```

```
public void setRegisterCriterias(criterias[] TTK91TaskCriteria)
Luo uuden vectorin, ja asettaa sen sisällöksi taulukkona annetut kriteerit.
```

```
public void addMemoryCriteria(criteria TTK91TaskCriteria)
Lisää annetun kriteerin muisti-vectorin viimeiseksi.
```

```
public void setMemoryCriterias(criterias[] TTK91TaskCriteria)
Luo uuden vectorin, ja asettaa sen sisällöksi taulukkona annetut kriteerit.
```

```
public void addScreenOutputCriteria(criteria TTK91TaskCriteria)
Lisää annetun kriteerin crt-vectorin viimeiseksi.
```

```
public void setScreenOutputCriterias(criterias[] TTK91TaskCriteria)
Luo uuden vectorin, ja asettaa sen sisällöksi taulukkona annetut kriteerit.
```

```
public void addFileOutputCriteria(criteria TTK91TaskCriteria)
Lisää annetun kriteerin file-vectorin viimeiseksi.
```

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

Luo uuden vectorin, ja asettaa sen sisällöksi taulukkona annetut kriteerit.

```
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. Arvot on määritelty luokassa TTK91Contant.

```
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 [] getRegisterCriterias()
```

Hakee kaikki rekistereihin liittyvät kriteerit.

```
public TTK91TaskCriteria [] getMemoryCriterias()
```

Hakee kaikki muistiin liittyvät kriteerit.

```
public TTK91TaskCriteria [] getScreenOutputCriterias()
```

Hakee kaikki näytön tulosteisiin liittyvät kriteerit.

```
public TTK91TaskCriteria [] getFileOutputCriterias()
```

Hakee kaikki tiedoston tulosteisiin liittyvät kriteerit.

```
public String getExampleCode()
```

Hakee mallivastauksen.

```
public String getTaskDescription()
```

Hakee tehtävänannon.

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

Hakee opiskelijalle näytettävät syötteen.

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

Hakee salaiset testisyötteen.

```
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.

3.4.5 TTK91TaskFeedback

Kääreluokka joka sisältää kaiken tehtävään liittyvän palautteen.

```
public void setAcceptedSizeFeedback(String positive, String negative,
String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen ohjelman maksimikoosta.

```
public void setOptimalSizeFeedback(String positive, String negative,
String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen ohjelman optimaalisesta koosta.

```
public void setRequiredCommandsFeedback(String positive, String negative,
String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen ohjelman vaadituista käskyistä.

```
public void setForbiddenCommandsFeedback(String positive, String negative, String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen ohjelman kielletyistä käskyistä.

```
public void setRegisterFeedback(String positive, String negative, String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen rekistereistä.

```
public void setMemoryFeedback(String positive, String negative, String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen muistipaikoista ja muuttujista.

```
public void setMemoryReferencesFeedback(String positive, String negative, String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen muistiviitteiden määräämistä.

```
public void setScreenOutputFeedback(String positive, String negative, String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen näytön tulosteista.

```
public void setFileOutputFeedback(String positive, String negative, String quality)
```

Asettaa positiivisen, negatiivisen ja laadullisen palautteen tiedoston tulosteista.

```
public String getAcceptedSizeFeedbackPositive()
```

Antaa positiivisen palautteen maksimikoosta.

```
public String getAcceptedSizeFeedbackNegative()
```

Antaa negatiivisen palautteen maksimikoosta.

```
public String getAcceptedSizeFeedbackQuality()
```

Antaa laadullisen palautteen maksimikoosta.

```
public String getOptimalSizeFeedbackPositive()
```

Antaa positiivisen palautteen optimaalisesta koosta.

```
public String getOptimalSizeFeedbackNegative()
```

Antaa negatiivisen palautteen optimaalisesta koosta.

```
public String getOptimalSizeFeedbackQuality()
```

Antaa laadullisen palautteen optimaalisesta koosta.

```
public String getRequiredCommandsFeedbackPositive()
```

Antaa positiivisen palautteen vaadituista käskyistä.

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

`public String getRequiredCommandsFeedbackQuality()`
Antaa laadullisen palautteen vaadituista käskyistä.

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

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

`public String getForbiddenCommandsFeedbackQuality()`
Antaa laadullisen palautteen kielletyistä käskyistä.

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

`public String getRegisterFeedbackNegative()`
Antaa negatiivisen palautteen rekistereistä.

`public String getRegisterFeedbackQuality()`
Antaa laadullisen palautteen rekistereistä.

`public String getMemoryFeedbackPositive()`
Antaa positiivisen palautteen muistipaikoista ja muuttujista.

`public String getMemoryFeedbackNegative()`
Antaa negatiivisen palautteen muistipaikoista ja muuttujista.

`public String getMemoryFeedbackQuality()`
Antaa laadullisen palautteen muistipaikoista ja muuttujista.

`public String getMemoryReferencesFeedbackPositive()`
Antaa positiivisen palautteen muistiviitteiden määrästä.

`public String getMemoryReferencesFeedbackNegative()`
Antaa negatiivisen palautteen muistiviitteiden määrästä.

`public String getMemoryReferencesFeedbackQuality()`
Antaa laadullisen palautteen muistiviitteiden määrästä.

`public String getScreenOutputFeedbackPositive()`
Antaa positiivisen palautteen näytön tulosteista.

`public String getScreenOutputFeedbackNegative()`
Antaa negatiivisen palautteen näytön tulosteista.

```
public String getScreenOutputFeedbackQuality()
Antaa laadullisen palautteen näytön tulosteista.
```

```
public String getFileOutputFeedbackPositive()
Antaa positiivisen palautteen tiedoston tulosteista.
```

```
public String getFileOutputFeedbackNegative()
Antaa negatiivisen palautteen tiedoston tulosteista.
```

```
public String getFileOutputFeedbackQuality()
Antaa laadullisen palautteen tiedoston tulosteista.
```

3.4.6 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.

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 13 ja 14.

4.1 Composer – tehtävän luonti ja muokkaus

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

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 13: Käyttöliittymähahmotelma: staattisen tehtävän luonti

Opiskelijalle annettavat palautteet			
Kriteeri	Kriteerit täyttyvät	Kriteerit eivät täyty	Laatu ei riitävän hyvä
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"/>	<input type="text"/>
Ohjelmassa kielletyt käskyt	<input type="text"/>	<input type="text"/>	<input type="text"/>
Rekisterien sisältö	<input type="text"/>	<input type="text"/>	<input type="text"/>
Muistipaikkojen ja muuttujien sisältö	<input type="text"/>	<input type="text"/>	<input type="text"/>
Muistiviitteiden määrä	<input type="text"/>	<input type="text"/>	
Tulosteet näytölle	<input type="text"/>	<input type="text"/>	<input type="text"/>
Tulosteet tiedostoon	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="text" value="Luo ja tallenna tehtävä"/>		

Kuva 14: Käyttöliittymähahmotelma: palautemäärittely

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 eroteltava 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

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.

Dynaamista tehtävää ei toteutettu Koskelo-projektissa kiireisen aikataulun takia.

4.2 Displayer – tehtävän ratkaisu ja palaute

Displayer huolehtii käyttöliittymän rakentamisesta. Sillä itsellään ei ole käyttäjälle näkyvää käyttöliittymää. Displayer toimii eri tehtävätyyppien osalta hieman eri tavoin. Esimerkiksi Fillin-tehtävien näytössä Displayer paloittelee malliratkaisun kolmeen osaan, ja piilottaa siitä keskimmäisen osan, jonka tilalle se luo tekstikentän johon opiskelija syöttää ratkaisunsa.

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 CommonAnalyserin näkyvyysmäärittely muutettu

Luokka `CommonAnalyser`, mikä on sijoitettu pakkaukseen `fi.hy.eassari.showtask.trainer`, oli alunperin esitely seuraavasti:

```
abstract class CommonAnalyser implements AnalyserInterface
```

Koska näkyvyysmäärittelyä ei erikseen mainittu, sai `CommonAnalyser` oletusnäkyvyyden (pakkausnäkyvyys). Koska Koskelo-ryhmän komponentit sijoittuvat omaan pakkausrakenteeseensa, ei `CommonAnalyseriin` ollut pääsyä. Asiakkaalta saatiin lupa muuttaa `CommonAnalyserin` näkyvyydeksi `public`, mikä korjaa ongelman. `CommonAnalyserin` uusi esittely on siis seuraava:

```
public abstract class CommonAnalyser implements AnalyserInterface
```

5.2 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:

STATIC_TTK91 = staticttk91task Staattinen ohjelmointitehtävä.

FILLIN_TTK91 = fillinttk91task Täydennystehtävä.

DYNAMIC_TTK91 = 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 ('statictk91task', 'exampleCode', 'TEXT', 'N', 'STORED');
```

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

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

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

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

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

```
insert into taskattributes values ('statictk91task', 'requiredCommandsQualityFeed-
back', 'TEXT', 'Y', 'STORED');
```

```
insert into taskattributes values ('statictk91task', 'forbiddenCommandsQuality-
Feedback', 'TEXT', 'Y', 'STORED');
```

Sivujen kieliversiot lisätty kantaan seuraavasti:

Staattisen tehtävämäärittelyn tekstit

```
insert into attributevalues values ('D', 'statictk91taskcomposer', 'exampleCodeHea-
der', 'FI', 'C', 'Ohjelman malliratkaisu');
```

```
insert into attributevalues values ('D', 'statictk91taskcomposer', 'taskDescription-
Header', 'FI', 'C', 'Tehtävänanto');
```

```
insert into attributevalues values ('D', 'statictk91taskcomposer', 'publicInputHea-
der', 'FI', 'C', 'Opiskelijalle näytettävät syötteet');
```

```
insert into attributevalues values ('D', 'statictk91taskcomposer', 'hiddenInputHea-
der', 'FI', 'C', 'Salaiset syötteet');
```

```
insert into attributevalues values ('D', 'statictk91taskcomposer', 'compareMet-
hod1Header', 'FI', 'C', 'Vertailu malliratkaisun suorituksen perusteella');
```

```
insert into attributevalues values ('D', 'statictk91taskcomposer', 'compareMet-
hod2Header', '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', 'submitButtonText', 'FI', 'C', 'Tehtävän luonti osa 2');
```

Tehtävämäärittelyn palautteenkeruulomakkeen tekstit ja virheilmoitukset

```
insert into attributevalues values('D', 'ttk91syntaxchecker', 'feedbacktitle', 'FI', 'C', 'Opiskelijalle annettavat palautteet');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker', 'criteriafilledtitle', 'FI', 'C', 'Kriteerien vaatimukset täyttyvät');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker', 'criterianotfilledtitle', 'FI', 'C', 'Kriteerien vaatimukset eivät täyty');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker', 'criteriaqualitytitle', 'FI', 'C', 'Laadulliset kriteerit täyttyvät');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91memoryvaluesyntaxerror', 'FI', 'C', 'Muistikriteerien syntaksi on väärin');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91registervaluesyntaxerror', 'FI', 'C', 'Rekisterikriteerien syntaksi on väärin');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91taskdescriptionsyntaxerror', 'FI', 'C', 'Tehtävänannon syntaksi on väärin');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91publicinputsyntaxerror', 'FI', 'C', 'Opiskelijalle näytettävien syötteiden syn-
taksi on väärin');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91hiddeninputsyntaxerror', 'FI', 'C', 'Opiskelijalta piilotettujen syötteiden syn-
taksi on väärin');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91examplecodesyntaxerror', 'FI', 'C', 'Malliratkaisun syntaksi on virheellinen');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91examplecodefillinsyntaxerror', 'FI', 'C', 'Malliratkaisuun on merkitty täyttö-
kohta virheellisesti');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91maxcommandssyntaxerror', 'FI', 'C', 'Ohjelman maksimikäskymäärä on ilmoi-
tettu virheellisesti');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91acceptedsizesyntaxerror', 'FI', 'C', 'Ohjelman hyväksymisen maksimikoko on
ilmoitettu virheellisesti');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91optimalsizesyntaxerror', 'FI', 'C', 'Ohjelman ihannekoko on ilmoitettu virheel-
lisesti');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91comparemethodsyntaxerror', 'FI', 'C', 'Ohjelman tarkistuksen vertailumetodi
on ilmoitettu virheellisesti');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91requiredcommandssyntaxerror', 'FI', 'C', 'Vaadittujen konekäskyjen syntaksi
on väärin. Tarkista, että kaikki määrittelemäsi käskyt ovat TTK91-käskyjä');

insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91forbiddencommandssyntaxerror', 'FI', 'C', 'Kiellettyjen konekäskyjen syntaksi
on väärin. Tarkista, että kaikki määrittelemäsi käskyt ovat TTK91-käskyjä');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91missingexamplecoderror', 'FI', 'C', 'Simuloitava mallivastaus puuttuu');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91memoryreferencessyntaxerror', 'FI', 'C', 'Muistiviitteiden määrä on ilmoitettu
virheellisesti');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91screenoutputsyntaxerror', 'FI', 'C', 'Näyttötulosteiden syntaksi on väärin');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker',
'ttk91fileoutputsyntaxerror', 'FI', 'C', 'Tiedostotulosteiden syntaksi on väärin');
```

```
insert into attributevalues values('D', 'ttk91syntaxchecker',
'submitbutton', 'FI', 'C', 'Tallenna');
```

eAssarin yleisen tehtävänmäärittelyn tekstit

```
insert into attributevalues values ('D', 'teachertaskdef', 'staticttk91tasktypelabel',
'FI', 'C', 'TTK91 ohjelmointitehtävä');
```

```
insert into attributevalues values ('D', 'teachertaskdef', 'fillinttk91tasktypelabel',
'FI', 'C', 'TTK91 täydennystehtävä');
```

Tehtävän näyttämisen syöte-teksti

```
insert into attributevalues values('D', 'staticttk91taskdisplayer', 'inputHeader', 'FI',
'C', 'Syötteet');
```

```
insert into attributevalues values('D', 'staticttk91taskdisplayer', 'inputHeader', 'EN',
'C', 'Input');
```

Palautteen muodostamisessa käytetyt tekstit

```
insert into attributevalues values('A', 'ttk91feedbackcomposer', 'criteriaLabel', 'FI',
'C', 'Kriteerit');
```

```
insert into attributevalues values('A', 'ttk91feedbackcomposer', 'criteriaLabel', 'EN',
'C', 'Criteria');
```

```
insert into attributevalues values('A', 'ttk91feedbackcomposer', 'feedbackLabel',
'FI', 'C', 'Palaute');
```

```
insert into attributevalues values('A', 'ttk91feedbackcomposer', 'feedbackLabel',
'EN', 'C', 'Feedback');
```

```
insert into attributevalues values('A', 'ttk91feedbackcomposer', 'qualityLabel', 'FI',
'C', 'Laadullinen palaute');
```

```
insert into attributevalues values('A', 'ttk91feedbackcomposer', 'qualityLabel', 'EN', 'C', 'Qualitative feedback');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'registervaluesLabel', 'FI', 'C', 'Rekisterit');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'registervaluesLabel', 'EN', 'C', 'Registers');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'statisticsLabel', 'FI', 'C', 'Statistiikka');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'statisticsLabel', 'EN', 'C', 'Statistics');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'outputLabel', 'FI', 'C', 'Tulosteet');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'outputLabel', 'EN', 'C', 'Output');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'memoryReference', 'FI', 'C', 'Muistiviittauksia yhteensä');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'memoryReference', 'EN', 'C', 'Total memory references');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'stackSize', 'FI', 'C', 'Pinon suurin koko');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'stackSize', 'EN', 'C', 'Stack max size');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'codeSegment', 'FI', 'C', 'Koodisegmentin koko');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'codeSegment', 'EN', 'C', 'Code segment size');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'dataSegment', 'FI', 'C', 'Datasegmentin koko');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'dataSegment', 'EN', 'C', 'Data segment size');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'executedCommands', 'FI', 'C', 'Suoritettuja käskyjä');
```

```
insert into attributevalues values ('A', 'ttk91feedbackcomposer', 'executedCom-
mands', 'EN', 'C', 'Executed commands');
```

5.3 Lisäykset Titokoneeseen

Kaikki muutokset on mainittu koodissa kommentteilla.

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]