Software Architecture Group
Faculty of Informatics
University of Vienna
http://cs.univie.ac.at/swa

050132 Advanced Software Architecture (UE), Winter Semester 2014/15
Task 3

Task 3
Domain specific language with Xtext

Task-ID: XTX

General Remarks

- The **deadline** for this work is the 30.11.2014 which can also be found in Moodle (https://moodle.univie.ac.at/course/view.php?id=34717).
  The deadline is firm! No deadline extensions are given.
- You will have to present your solution – so be prepared
  (laptop, pdf of your presentation, running example, …)
- You can get up to 10 points, which have to be defended during presentation.
- This is an **individual work**. Consequently, group work is not allowed in this assignment.
- If you have problems, do not hesitate to contact the tutor (ase.tutor@swa.univie.ac.at) or us (ase@swa.univie.ac.at) via e-mail.

Submission Guidelines

- The guidelines proposed on https://swa.univie.ac.at/General_guidelines must **always** be satisfied.
- If guidelines are ignored,
  - an assessment of the tasks **cannot be guaranteed** because they are processed electronically.
  - one (1) point will be deducted from the assessment of the task for each unsatisfied guideline.
- Artifacts to upload into Moodle are (see guidelines for more information):
  One (1) ZIP archive containing:
  - 1 Eclipse project that contains your DSL
  - All automatically created Eclipse projects
  - 1 instance of your DSL!
  - Document about your assumptions (PDF)

Task Definition

Right at the Start

Please, read the guidelines at https://swa.univie.ac.at/General_guidelines, especially the part about naming conventions for your uploads! If you do not follow these guidelines, we will not be able to grade your work, because we will not be able to find your files any more (yes, it is that simple …).

Task 3a:

Create a domain specific language (DSL) for the scenario described below using Xtext! Your DSL should be easy to read and understand for people who do not have a computer science background. The instances for your DSL should be similar to the exemplary instance shown below.

Software Architecture Group
Faculty of Informatics
University of Vienna
http://cs.univie.ac.at/swa

050132 Advanced Software Architecture (UE), Winter Semester 2014/15
Task 3
Description of the music streaming library model:

An online streaming music library contains a collection of songs. Each song has a name, an artist, a distributor, a length, a price, and a genre. At least the following genres should exist in our DSL: POP, ROCK, HARDROCK, CLASSIC, FOLK.
Distributors have a unique name, an address, and account information.
Your DSL needs to allow the definition of playlists based on the songs in the library and based on other playlists.
This means that a playlist (which has a unique name) can contain an arbitrary number of songs, and can reuse other playlists by placing the name of the reused playlist somewhere on to the new playlist. It should also be possible to exclude some songs from the reused playlist. Please see the example instance in the Appendix for details.

Create your DSL from scratch. Have in mind that you will implicitly create a domain model.
Note: An automatically generated grammar will not be accepted! Although it is possible to automatically create a grammar from an ecore model, it will never fit the requirements for a user (like readability). Remark: It is very easy to distinguish a generated grammar from a manually created one – do it on your own!

Task 3b:

Create one instance of the DSL you created. This instance needs to contain at least three distributors and at least 10 songs, and three playlists from those three, two playlists have to contain at least three songs and the third playlist has to reuse at least one other playlist.

Task 3c:

Create an Xtend transformation that creates one HTML table per playlist that lists the songs in each playlist (pure HTML is sufficient – it is not necessary to add style information). In the HTML table only songs are listed (songs from referenced playlists are also listed). Furthermore your transformation should also create an extra HTML table that lists the earnings per distributor over all existing playlists (sum up of the prices) sorted by the amount of earnings (descending).

Resources and Tipps:

1) Xtext documentation: http://www.eclipse.org/Xtext/documentation.html
2) Extended 15 minute tutorial on writing a code generator with xtend: http://www.eclipse.org/Xtext/documentation.html#DomainmodelNextSteps
3) Xtend documentation: http://www.eclipse.org/xtend/documentation.html
4) Create your instances with your generated DSL editor. You will find the created data in your workspace runtime-EclipseXtext (or runtime-EclipseApplication).

Feel free to make your own assumptions. Mention them in your document.
If you have questions about the requirements, feel free to discuss them with your colleagues and us using the moodle forum.

Software Architecture Group
Faculty of Informatics
University of Vienna
http://cs.univie.ac.at/swa

050132 Advanced Software Architecture (UE), Winter Semester 2014/15
Task 3
Checklist
Before uploading your solution, ensure the following checkpoints can be marked positive:

- I followed the guidelines and conventions from https://swa.univie.ac.at/General_guidelines
- I created an Xtext DSL for the described model.
- I generated an editor for my Xtext grammar.
- I created an instance of the DSL and included it in my ZIP file.

Appendix – Example of a DSL Instance

```
Distributors:
	UniversalMusic
	Address: "56 Some Street, 12345 New York, NY, US"
	Account Information:
		IBAN: "US 12 123 123 123"
		BIC: "SOMEUSX1"

	SonyMusic
	Address: "1 Sony Street, 2345 Tokia, JP"
	Account Information:
		IBAN: "JP 12 34 43 55 93"
		BIC: "SOMEJPJ1"

Library:
	Come_as_you_are
	sung by "Nirvana"
	produced by UniversalMusic
	length: 7:23
	genre:ROCK
	price: 7.75

	Bohemian_Rapsody
	sung by "Queen"
	produced by UniversalMusic
	length: 5:32
	genre: ROCK
	price: 3.65

	Get_lucky
	sung by "Daft Punk"
	produced by SonyMusic
	length: 3:01
	genre: OTHER
	price: 4.32

	Song2
	sung by "Blur"
	produced by UniversalMusic
	length: 4:23
	genre: ROCK
	price: 3.45
```

Software Architecture Group
Faculty of Informatics
University of Vienna
http://cs.univie.ac.at/swa

050132 Advanced Software Architecture (UE), Winter Semester 2014/15

Task 3

```
      Roar
      sung by "Katie Perry"
      produced by SonyMusic
      length: 2:35
      genre: POP
      price: 4.67

Playlists:
      SimpleList consists of
      Bohemian_Rapsody,
      Come_as_you_are,
      Get_lucky

      ComplexList consists of
      Song2,
      Playlist SimpleList without Get_lucky,
      Roar

      RockList consists of
      Song2,
      Bohemian_Rapsody,
      Come_as_you_are
```