Laboration 1b: En lexikonmodul

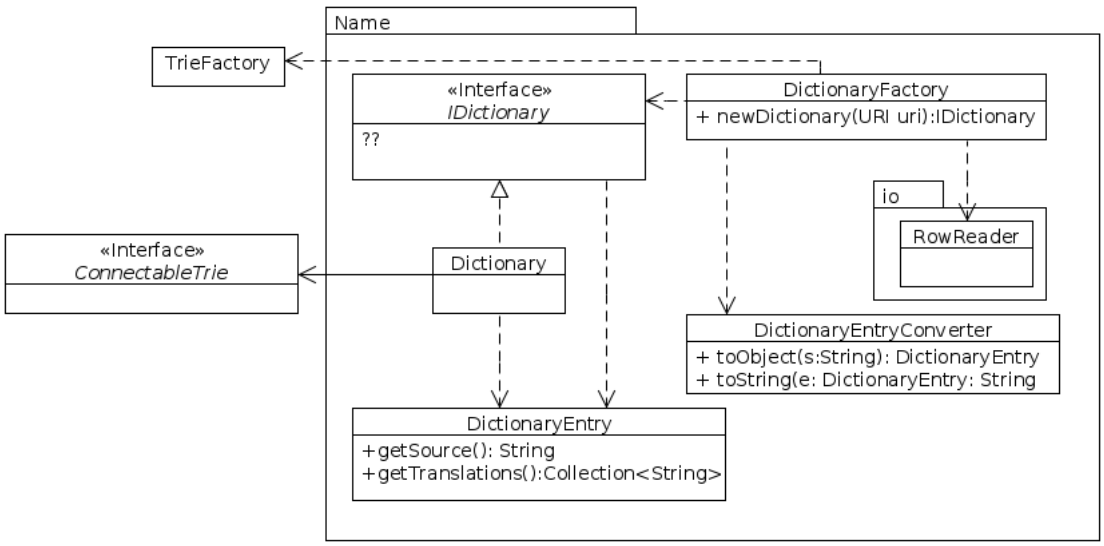
1 Syfte

Fortsättning på laboration 1a och mer OO-design.

2 Uppgift

Ni skall implementera Dictionary-modulen, se lab 1a. Modulen använder typer från Trie-projektet. Ni måste därför lägga till ett beroende från detta projekt till Trie-projektet. Använd "Build Path" i Eclipse.

3 Designmodell



Figur 1: Designmodell för Dictionary.

- Dictionary är implementationen av gränssnittet IDictionary som Translator senare skall använda sig av. Använder sig av IConnectableTrie.

- DictionaryEntry är en klass som håller ett ord och alla översättningar. DictionaryEntryConverter omvandlar mellan strängar och DictionaryEntry.
- Paketet io innehåller en klass för att läsa rader från en textfil, RowReader (skall egentligen ligga i egen modul).
- DictionaryFactory är en fabrik för Dictionary-objekt. Fabriken använder RowReader, DictionaryEntryConverter m.m. för att skapa ett Dictionary (returneras som gränssnitts-typen).

4 Problem

Tyvärr upptäcker vi i detta läge att Trie-modulens gränssnitt innehåller (med avsikt) en grov felaktighet. Dictionary kommer att kunna skaffa sig alldeles för mycket information om ConnectableTrie! Vad är problemet? Åtgärda!

Amn. Har vi en robust design skall detta inte leda till några större problem.

5 Data

Det finns textfiler som innehåller översättningar mellan Svenska och Engelska (US). Se katalogen dict.

6 Felhantering

Ingen felhantering här, felen skickas vidare till Translator, använd throws.

7 Implementation

Implementera de delar som fattas för att få ett fungerade lexikon. Det finns ett påbörjat Eclipse-projekt på kursidan.

- Du får själv bestämma vilka metoder som skall finnas i IDictionary. Tänk utifrån användarens perspektiv. Vad kommer Translator-modulen att behöva?
- Låt Dictionary använda två IConnectableTrie för att utföra själva arbetet. Klassen implementerar IDictionary. Klassen Dictionary kan ha metoder som används internt i modulen, dessa ingår inte i gränssnittet.
- DictionaryFactory fungerar på samma sätt som TrieFactory. En skillnad är dock att DictionaryFactory skall ta en URI som anger vart man skall hämta orden som skall finnas i lexikonet (från fil eller över nätet eller,..). Just nu så finns alla filer i dict-mappen. Behöver bara använda denna. Se eventuellt javadoc för klassen URI.

- Se till så att man kan sortera DictionaryEntry utifrån utgångsspråket. På så sätt kan användaren få en sorterad lista med ord och översättningar (översättningar ej sorterade). Sortering skall göras med Collections.sort(..).

8 Testning

Skapa nödvändiga (icke-triviala) tester parallellt med implementationen (en testklass bör räcka). Ni kan utgår från att RowReader är testad och klar.

9 Redovisning

Körningsgodkännande samt kodinspektion. Görs under laborationspassen. Se till att bli avprickad. Följande kommer att kontrolleras:

- Inspektion av reviderad Trie-modul.
- Allmän kodstil, indentering, krullparenteser, hårdkodade värden, ...
- JUnit-tester skall finnas och vara godkända.
- Inga varningar skall finnas.
- Alla klasser skall ha en klasskommentar (precis innan klassdeklarationen) som anger ev. klassinvarianter. Kommentera metoder som ev. påverkar invarianterna (visa att invarianter upprätthålls). Använd icke-muterbara klasser så långt det går
- Så lite som möjligt av modulen skall synas utåt. Så få associationer som möjligt mellan klasser.

Inlämningsdatum Se kurssidan.