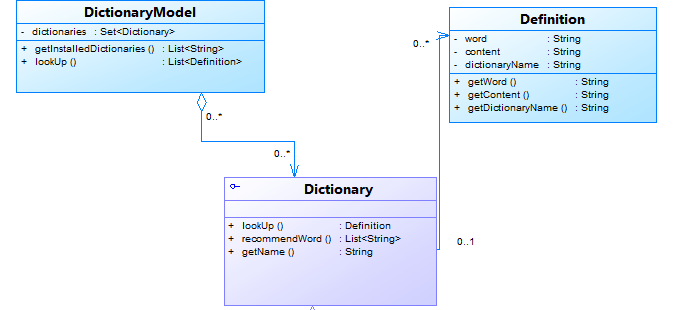
MEGADICT

# DICTIONARY MODEL

Package com.megadict.model



**Dictionary Interface** – common interface for all format dictionaries (currently we are only supporting DICT format, but StarDict is considering).

* **lookUp(String word)** : look up a word and retrieve it’s Definition.
* **recommendWord(String word)**: return a list of words (include itself if possible) that looks similar to the argument.
* **getName()**: returns the name of the Dictionary.

**DictionaryModel** – a planned model for Megadict.

* **getInstalledDictionaries()**: return a lists of installed dicitonaries (only returns name).
* **lookUp(String word):** returns a list of Definition(s) after calling all lookUp methods on installed dictionaries.

**Definition** – definition of a word

* **getWord()**: returns the (head)word of the definition.
* **getContent():** returns the content of the definion. Currently it returns the raw text retreive from .dict file.
* **getDictionaryName():** returns the name of dictionary it belongs to.

# DICT Implementation

Usage:

Code snippet below demonstrates how to use DICTDictionary:

String indexFile = "test\\indexFile.index";

String dictionaryFile = "test\\dict.dt.dz";

Dictionary dictionary = **new** DICTDictionary(indexFile, dictionaryFile);

// Get the name

System.out.println(dictionary.getName());

// Look up a word

Definition def = dictionary.lookUp("hello");

System.out.println(def.getContent());

// Get similar words

List<String> similarWords = dictionary.recommendWord("hello w");

System.out.println(similarWords);

Another Plan for API of DICTDictionary:

Dictionary dictionary = **new**

DICTDictionary.Builder(indexFile, dictfile).isZipped(**true**).isUseDatabase().build();

## Design

