programming techniques

HOMEWORK 5

DICTIONARY OF ENGLISH SYNONYMS

# Abstract

Consider a Dictionary of Synonyms for Romanian or English language. It is required to use Java Collection Framework Map for the implementation.

Define and implement a domain specific interface (populate / add / remove / copy / save / search, etc.). Consider the implementation of specific utility programs for dictionary processing. For example:

- Implement a method for checking dictionary consistency. A dictionary is consistent, if all words that are used for defining a certain word are also defined by the dictionary.

- Implement dictionary searching using “\*” (any string, including null) and “?” (one character). For example, you can search for *a?t*.

# Application Description

The given task has been interpreted in the form of a Dictionary of Synonyms. This Dictionary of Synonyms should provide users with the possibility of searching for words, adding new words to the thesaurus, removing unused words and listing all its content. All the information about the Dictionary’s state is saved in a file after the user closes the application.

* A **thesaurus** (plural "thesauri") is a reference work  that lists words grouped together according to similarity of meaning (containing synonyms and sometimes antonyms), in contrast to a dictionary, which provides definitions for words, and generally lists them in alphabetical order.The main purpose of such reference works is to help the user “to find the word, or words, by which [an] idea may be most fitly and aptly expressed” – to quote Peter Roget, architect of the best known thesaurus in the English language.
* In this documentation, we will sometimes referr to the *Dictionary of Synonyms* as a dictionary and sometimes, for not repeating the same word, as a *thesaurus* or *collection of words*.
* The data was obtained from published dictionaries and reflects the diversity of the language within these sources. One way of viewing the output of the program is to look at a list of the semantic relations for a given entry, where each line of the list represents a different *component*. In this application, there will be no limit of requests per day.

# General Overview

One major advantage of having a dictionary of synonyms on the computer is the possibility of searching and finding quickly the word which is searched for.The user can search for a word by writing the entire word in the search column,or he can search it by using the “\*” and “?” characters.The “\*” character is used in case the user wants to search for a word,but skips writing multiple consecutive characters,replacing them with “\*”.The “?” is used when the user wants to search for a word and skips writing a letter.The user has also the possibility to add words and their synonyms,or to remove specific words,or again,their synonyms.

# Concepts related to this project

In this project I have used my knowledge from the previous assignments.I have therefore used the concepts related to Design By Contract,meaning invariants ,preconditions and postconditions implemented as comments in the DictionaryProc interface and tested with assert in the Dictionary class.I have also used the function isWellFormed to check the consistentcy of the Dictionary.If a word of the dictionary contains a synonym,then the synonym must be a word itself in the dictionary and have as a synonym out previously discussed word.

I have also used the Java Collection Framework for implementing a HashMap that would store a String as the key value,and a ArrayList of Strings as synonyms. Hash table based implementation of the Map interface. This implementation provides all of the optional map operations, and permits null values and the null key. (The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls.) This class makes no guarantees as to the order of the map; in particular, it does not guarantee that the order will remain constant over time.

This implementation provides constant-time performance for the basic operations (get and put), assuming the hash function disperses the elements properly among the buckets. Iteration over collection views requires time proportional to the "capacity" of the HashMap instance (the number of buckets) plus its size (the number of key-value mappings). Thus, it's very important not to set the initial capacity too high (or the load factor too low) if iteration performance is important.

For the new learned concepts,the most important are the usage of 2 disctinct Design Patterns ,in my case the MVC and the Singleton design Pattern. Singleton Design Pattern is part of creation design patterns and the strategy of singleton design pattern is oriented towards creation of objects. Singleton design pattern ensures that only one instance is created for a class..The key ways in implementing the Singleton class were the usage of a private access modifier to create the constructor of the dictionary class,and the creation of a instance of the Dictionary class withing the class. This ensures that no other code in your application will be able to create an instance of this class. This enforces the requirement that there ever only be one instance of this class. The instance was returned to the main class and its purpose was to represent the whole class,therefore the static access modifoer was used to define the return method.

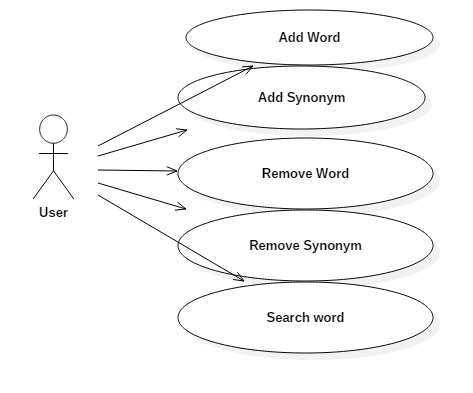
The second design pattern used was MVC. The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the **view**, and the **controller**. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

**Model:** The Model component corresponds to all the data related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

**View:** The View component is used for all the UI logic of the application. For example, the Customer view would include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

**Controller:** Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller would handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller would be used to view the Customer data.

# Use cases



**1. Add word**

**Title:** *Add word*

**Resume:** The user inserts a new word into the application by filling in data in the two text fields from the graphical user interface corresponding to the word.

**Actors:** User

**2. Add word**

**Title:** *Add word*

**Resume:** The user inserts a new word into the application by filling in data in the two text fields from the graphical user interface corresponding to the word and to its synonym(s).

**Actors:** User

**3. Delete synonym**

**Title:** *Delete synonym*

**Resume:** The user deletes a synonym from the dictionary application by inputing the entry he wants to delete, clicking the delete button, and confirming the removal in the shown dialog.

**Actors:** User

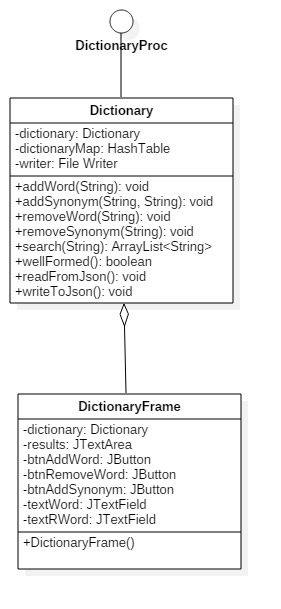
**4. Search synonym**

**Title:** *Search*

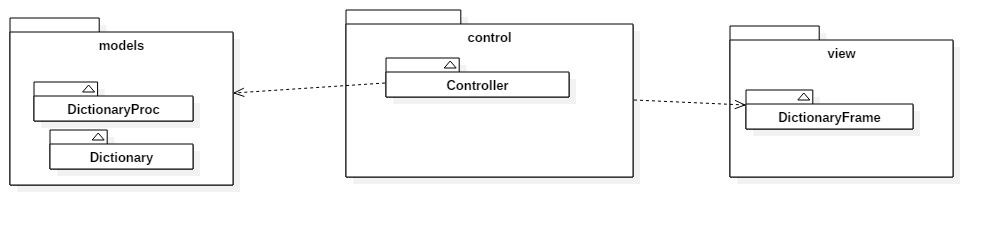
**Resume:** The user filters the words displayed in the list, by typing a search input in the search field. The result list is updated when pressing enter or clicking on search. Then, he can return to the home screen by pressing the transformed-search button that now says “Home”.

**Actors:** User

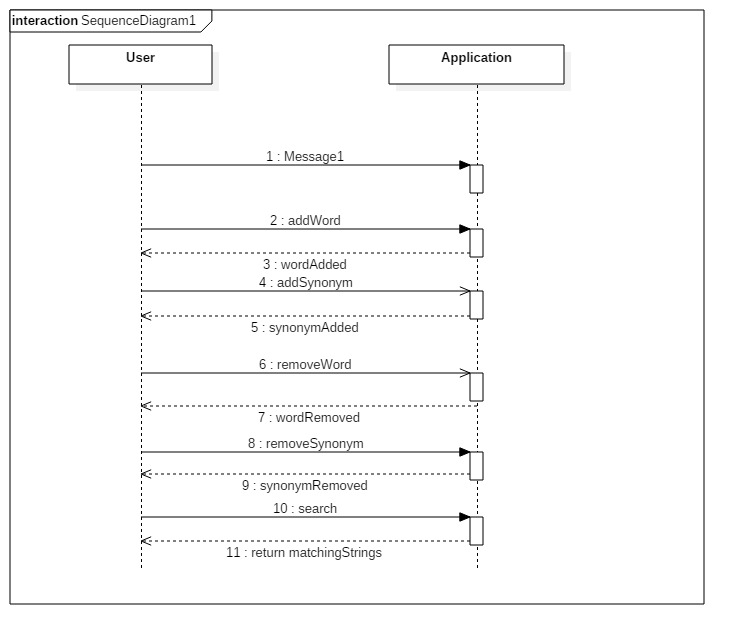
# Class Diagram



# Package Diagram



# Sequence Diagram



# JSON

A JSON file was used to keep the data.

JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent data exchange format that is easy for humans and machines to read and write. JSON can represent two structured types: objects and arrays. An object is an unordered collection of zero or more name/value pairs. An array is an ordered sequence of zero or more values. The values can be strings, numbers, booleans, null, and these two structured types.

This is an example from Wikipedia that shows the JSON representation of an object that describes a person. The object has string values for first name and last name, a number value for age, an object value representing the person's address, and an array value of phone number objects.

{

"firstName": "John",

"lastName": "Smith",

"age": 25,

"address": {

"streetAddress": "21 2nd Street",

"city": "New York",

"state": "NY",

"postalCode": 10021

},

"phoneNumbers": [

{

"type": "home",

"number": "212 555-1234"

},

{

"type": "fax",

"number": "646 555-4567"

}

]

}

Here are the functions that implement the writing and the reading from the JSON. In our case, it’s related to reading and writing data to and from the HashMap into the JSON file *dictionary.txt*.

# Conclusions

By means of using JSON files and a friendly graphical user interface, a user-friendly application which simulates the functionality of a Romanian synonyms dictionary has been developed. However, there are further improvements that can be done, in order to adapt the application to real-world situations. Moreover, this small application can be modified into an applet in order to be published onto a website.

Future improvements:

* Populating with existing verified dictionaries
* Improving the search with more matching algorithms
* Sounds for pronunciation of words
* Connecting the files to the web

# Bibliography

jSON

http://www.json.org/ https://sites.google.com/site/gson/gson-user-guide http://wiki.fasterxml.com/JacksonInFiveMinutes

Design Patterns

http://www.tutorialspoint.com/design\_pattern/

<http://docs.oracle.com/javase/6/docs/api/java/util/Hashtable.html>

<http://stackoverflow.com/questions/1738963/is-hashmap-in-java-collision-safe>

<http://docs.oracle.com/javase/6/docs/api/java/util/AbstractList.html#equals(java.lang.Object)>

<http://docs.oracle.com/javase/6/docs/api/java/util/ArrayList.html>

<http://docs.oracle.com/javase/6/docs/api/java/util/Iterator.html>

<http://docs.oracle.com/javase/6/docs/api/java/util/Set.html>

<http://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html>

<http://users.utcluj.ro/~jim/OOPE/>