

**PROGRAMMING TECHNIQUES**

Homework 5

**Student:Nicolescu Alexandru**

**Group:** 30422

**Table of Contents**

**1. Objective 3**

**2. Problem analysis, modeling, scenarios, use cases 3**

**2.1. Problem Analysis 3**

**2.2. Modeling 4**

**2.3. Scenario and use cases 4**

**3. Design 5**

**3.1. UML Diagram 5**

**3.2. Data Structures 6**

**3.3. Class Projection 6**

**3.4. Design Patterns 8**

**4. Implementation and use of cases 9**

**5. Results 10**

**6. Conclusion and possible updates 10**

**7. References 10**

**Documentation Dictionary implementation**

1. Objective

This homework’s aim is to use elements of object-oriented programming to implement a Dictionary of synonyms.

1. Problem analysis, scenarios, use cases

*Consider the implementation of one of the following:*

*a) A dictionary of Romanian language or a dictionary of English language or*

*b) A dictionary of synonyms (thesaurus) 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\*.*

*Use the above examples to warm up your imagination.*

2.1 Problem analysis

If we take a closer look to the problem of dictionary management we shall find that it is a more complex problem that it may seem at first glance.

First of all we need to find an OOP style way to store the data that comes from the serializable file. I have used an object for each word and synonyms. Therefore we have : String, Set<String> and Map<String, Set<String>>.

Another problem concerning the word processing would be the way the user gives us the data and the way we tell him the result of the processing. After doing some research on the internet and looking at online applications that provide these services I have reached the conclusion that the best way in terms of displaying data which comes from tables is using naturally a line sorting of the words which is put in the Graphic User Interface along with the data input section.

Further on we shall analyze all the aspects which needed to be managed in order for the right functioning of this project.

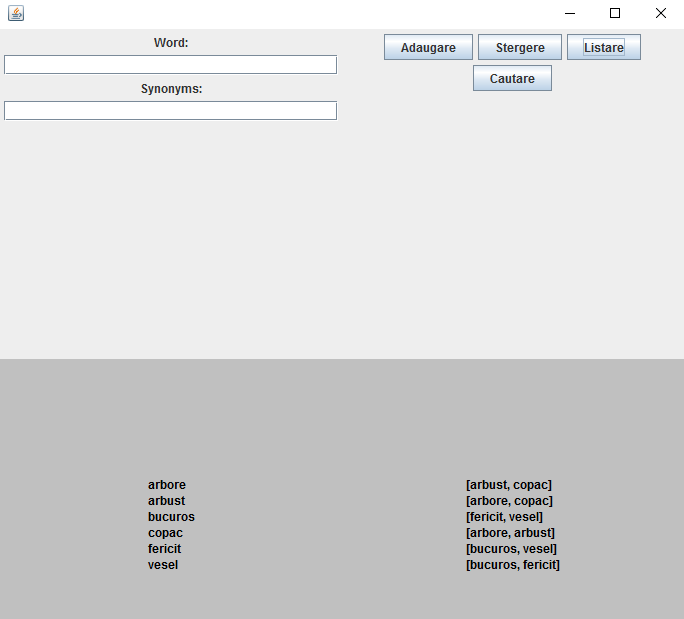
* 1. Modeling

The implementation of this project is the same one as the one presented in the section 2.1. The “assig5” package which stands for Assignemnt 5 has all the logic of storing and displaying data while the “GUI” package naturally only makes the application-user connection using a very smooth and efficient data inputting.

2.3 Scenarios and use cases

The scenarios were already mentioned, but I will present the details here. Firstly I thought about how the data will be managed by the application and my idea was: GUI for reading the data -> Dictionary for processing the words -> Serialization classes to store the new words. I followed this plan and I did not have any unexpected surprise when it comes to the implementation and the structure of the program.

The use cases are strictly dependent on the user, and finally I order to make the application as user friendly as possible I decided to implement the following user interface:



From now on I will present the functionality of the application:

-“Adaugare”: The user needs to input a word a synonym. If that word exists already, it will be added to that group.

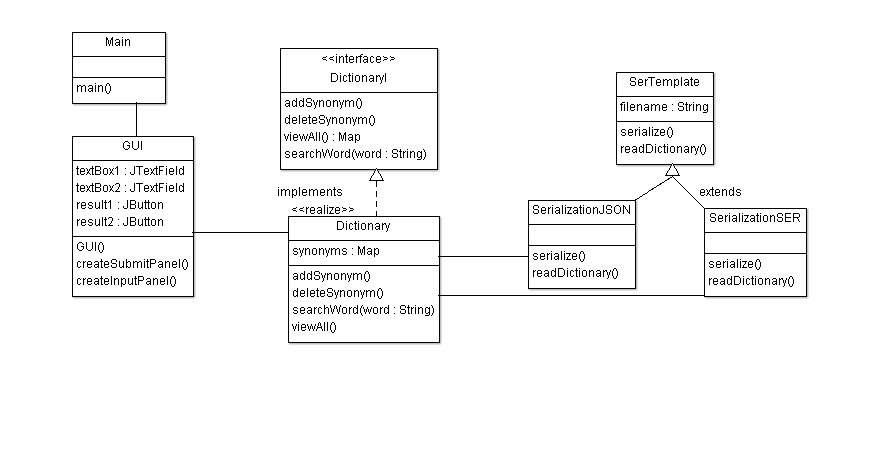
-“Listare”: Displays all the synonyms

-“Cautare”: Displays the synonyms of the word in the first textbox.

-“Stergere”: It deletes the word from the textbox.

1. Design

3.1 UML Diagram



The UML diagram is a class diagram in which we can find the relationship between classes and also the elements that the specified class contains.

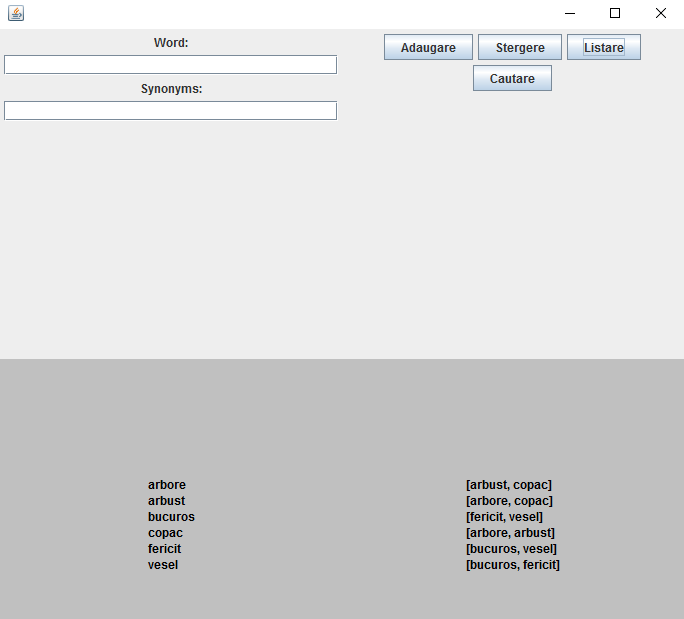
One could observe that for joining the classes I used several types of relationships. Between Dictionary and DictionaryI there is a implements relationship, extends between serialization classes and association between the others.

3.2 Data structures

In this application I have used various data types including JButtons, JTextFields, JPanels, JFrame for the GUI package and the classic int, String, Boolean for regular operations.

3.3 The interface

The interface is made out of 2 JPanels which are included in the JFrame in a GridLayout(1,2). We shall describe each JPanel in order to understand how the user interface is created.



This photo represents the input panel which is used in order for the user to enter the command data and/or select the command. Some of the data can be selected from the table. There is also a news button which is located on the button of the panel and it displays all the information for using this application and if the case, the errors resulted out of entering the wrong kind of data, therefore the use of this application is easy and does not need any other extra information.

3.4 Design Patterns

As the Assignment was to implement 2 Design Patterns I will present the ones that I have chosen:

1. Iterator pattern

Iterator<String> iterator = arr.iterator();

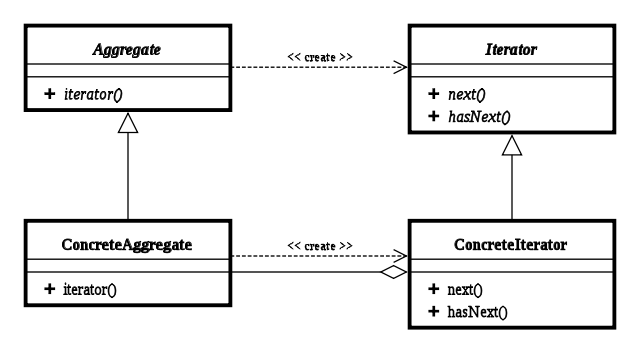
**while** (iterator.hasNext()) {

values.add(iterator.next());

}

In this part of the code I have used Iterator Pattern to iterate through a JSONArray structure. I will present the definition of Iterator and its class diagram:

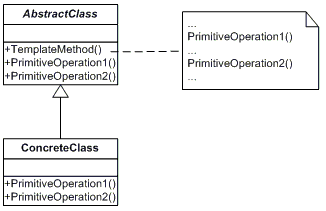
In object-oriented programming, the iterator pattern is a design pattern in which an iterator is used to traverse a container and access the container's elements. The iterator pattern decouples algorithms from containers; in some cases, algorithms are necessarily container-specific and thus cannot be decoupled.



1. Template Pattern

Template Pattern is used in the Serialization Classes using an abstract class in order to make a template for a serialization class as for further development.

In software engineering, the template method pattern is a behavioral design pattern that defines the program skeleton of an algorithm in a method, called template method, which defers some steps to subclasses. It lets one redefine certain steps of an algorithm without changing the algorithm's structure. This use of "template" is unrelated to C++ templates.

1. Implementation

In what the implementation is concerned this project was developed in Eclipse and it was only tested in this environment. However the program should maintain its portability. Concerning the code implementation I did not make use of laborious algorithms, but I have rather stayed faithful to the classical algorithms of computing polynomials learned in high school. However I have tried to implement my problem in a way that appears to me as being the most efficient one, this is why I have changed my model at first. Testing implies checking for any errors in the program or limitations of this program. Due to the fact that the program is rather simplistic, they are few errors that might generate this program to work wrong or to stop. These errors are mostly related to the interface or the database connection. I have assumed that the user reads the instructions from the interface and respects them, otherwise if he enters data with invalid format the program will probably generate some bugs and will stop. Hence this part with checking all the possible scenarios will be seen as future development.

1. Results

The application is an user friendly and useful application to perform basic create update delete and read operations on a sample created database. As the application is developed on a Java platform, it is highly portable and allows it to run on several operating systems (as long as they have the Java SDK installed). The application is straightforward an easy to understand and to use by any user who respects the instructions given in the interface and who has some basic knowledge of database storing, of course. Even though being limited, this application can be considered as being a helpful tool that can be used when dealing with such data storing situations.

1. Conclusions

All in all, the application works perfectly on the required operations and it is a user friendly interface which gives the users a simple and efficient answer to any problems related to a selected bank operation. As in possible updates I would add more output possibilities for security and I would increase the words number in the database.

1. References
2. <http://stackoverflow.com/questions/443499/convert-json-to-map>
3. http://www.tutorialspoint.com/design\_pattern/