Universitatea Tehnica din Cluj-Napoca Facultatea de Automatica si Calculatoare

# Dictionar RO - EN, EN - RO

Documentatie – tema 5 –

Czako Zoltan

Grupa: 30225

## Contents

1.Obiectivul temei	3
2. Analiza problemei , modelare , scenarii , cazuri de utilizare	3
3. Proiectare	3
3.1 Proiectare clase	4
3.2 Diagrama UML	6
3.5 Interfața grafică	7
4. Testare	11
5. Rezultate	12
6. Concluzii	12
7. Bibiliografie	13

#### 1. Objectivul temei:

Consider the implementation of one of the following:

- i. A dictionary of Romanian language or a dictionary of English language or
- ii. 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. Analiza problemei, modelare, scenarii, cazuri de utilizare

Aceasta problema apare si in viata reala, de multe ori ne cu situatii in care trebuie sa comunicam cu oameni vorbitori de limbi straine, iar o modalitate de a comunica cu aceste oameni este folosirea unei limbi comune. In zilele noastre o limba foarte raspandita e limba engleza, aproape toata lumea cunoaste aceasta limba, copii invata inca incepand cu gradinita sau cu clasa 1. Chiar daca suntem vorbitori buni a limbii engleze, fiind o limba straina, de multe ori uitam un cuvant si repede ar trebui sa gasim undeva acest cuvant. Aceasta aplicatie e perfecta pentru situatii de geznul acesta, fiind vorba de un dictionar englez - roman, roman – englez, orice cuvant vreb sa cautam, nu trebuie sa facem altceva decat sa introducem cuvantul dorit in limba romana si ne da traducerea in limba engleza sau daca nu intelegem un cuvant, introducem cuvantul englez si ne traduce in limba romana. Chiar daca nu stim exact cum se scrie un cuvant, avem posibilitatea ca in loc de un caracter sau in loc de un sir de caractere sa introducem semnul ? sau semnul \*, iar dictionarul ne va da toate cuvintele care corespunde acestui sablon.

## 3. Proiectare (diagrama UML, structuri de date, proiectare clase, interfete, relații, packages, algoritmi, interfața utilizator)

Pentru rezolvarea cat mai corecta si pentru o aplicatie cat mai avansatsa, am adaugat cateva clase care ne ajuta la pre si post conditii, si clase care realizeaza o interfata intre program si utilizator, facand ca utilizatorul sa poate utiliza acest program foarte usor, fara sa aiba nevoie de cunostinte din domeniul programarii a calculatoarelor.

#### **Clasele existente:**

Dictionary – Contine un HashMap, care realizeaza o relatie de dependenta unica intre cuvintele si semnificatia lor, si care nu ne lasa sa introducem un cuvant de doua ori, adica sa avem chei duplicate, astfel contribuind la corectitudinea si la consistenta dictionarului. Aceasta clasa contine o serie de metode care realizeaza anumite operatii pe dictionarul nostru, operatii care pot fi folosite atat de utilizatorul programului cat si de ownerul acestui dictionar. Clasa Dictionary implementeaza interfata Idictionary, care contine metodele:

- public void initDictionary(String path);
- public void addWord(String ENword, String ROword);
- public void removeWord(String ENword, String ROword);
- public boolean checkWholeConsistency();
- public DefaultListModel genericSearch(String pattern) throws PatternSyntaxException;

#### Metodele existente:

- initDictionary: e o metoda pentru initializarea decitionarului. Umplerea dicitionarului cu cuvintele existente se face utilizand un fisier, care contine pe randuri separate cuvantul in limba engleza si semnificatia in limba romana, separate cu un spatiu pentru citirea si parsarea mai usoara a datelor.
- searchMeaning: e o metoda, care primeste un cuvant si returneaza semnificatia cuvantului in limba romana sau in limba engleza, in functie de limba in care introducem cuvantul
- addWord: e o metoda, care realizeaza adaugarea unui cuvant in dictionar, astfel incat proprietatea de consistenta a dictionarului sa fie pastrata
- simpleAddWord: e o metoda, care realizeaza adaugarea unui cuvant in dictionar, fara pastrarea proprietatii de consistenta a dictionarului
- removeWord: e o metoda, care realizeaza stergerea unui cuvant din dictionar, astfel incat proprietatea de consistenta a dictionarului sa fie pastrata
- checkConsistency: e o metode, care verifica pentru un cuvant daca este inrodus corect in dictionar, adica daca dupa introducerea cuvantului, dictionarul a ramas sau nu a ramas consistent
- checkWholeConsistency: e o metoda, care verifica consistenta intregului dictionar, parcurgand fiecare cuvant in parte din dictionar
- genericSearch: e o metoda, care realizeaza o cautare generica in dictionar, adica utilizatorul poate sa introduca in loc de un singur caracter semnul? sau in loc de un sir de caractere oarecare, de lungime variabila semnul\*. Metoda returneaza o lista cu cuvintele pentru care am avut matching. Pentru rezolvarea acestei probleme am folosit expresii regulate ( regex)

• showDictionary: e o metoda, care returneaza Un DefaultListModel cu toate cuvintele continute de dictionar si cu semnificati lor

BussinesLogic: e o clasa pentru verificarea corectitudinii datelor introduse, are rolul de a verifica conditiile in care dictionarul va functiona corect. De exemplu metoda verifyTextField verifica daca TextFieldul nu e gol in momentul apasarii butonului translate sau search in interfata grafica pentru utilizator sau in momentul apasarii butoanelor add, check etc. in cadrul interfetei grafice pentru control panelul dictionarului.

*EmptyStringException*: e o clasa de tip Exceptie, exceptie, care va fi aruncata in cazul in care exista un TextField gol gol in momentul apasarii butonului translate sau search in interfata grafica pentru utilizator sau in momentul apasarii butoanelor add, check etc. in cadrul interfetei grafice pentru control panelul dictionarului.

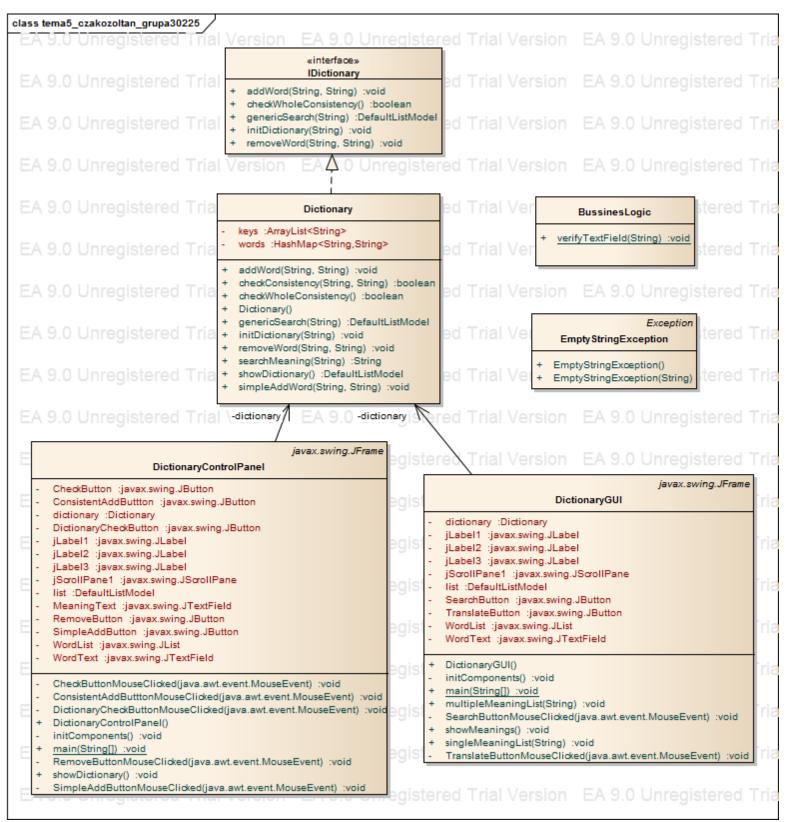
*DictionaryControlPanel:* e o clasa care realizeaza o interfata grafica pentru gestiunea datelor in interiorul dictionarului. Contine o serie de elemente Swing:

- TextField pentru cuvantul pe care vrem sa introduce sau pentru care vrem sa face un test de consistenta
- TextField pentru semnificati cuvantului pe care vrem sa introducem sau pentru care vrem sa face un test de consistenta
- Butonul SimpleAdd, pe care daca il apasam realizeaza introducerea cuvantului intre cuvintele continute de dictionar, fara pastrarea proprietatii de consistenta a dictionarului
- Butonul ConsistentAdd, pe care daca il apasam realizeaza introducerea cuvantului intre cuvintele continute de dictionar, cu pastrarea proprietatii de consistenta a dictionarului
- Butonul Remove, pe care daca il apasam va rezulta stergerea cuvantului ales din lista cuvintelor continute de dictionar
- List care contine o lista cu cuvintele si cu semnificatia lor din dictionar
- Labeluri, care fac etichetarea componentelor

DictionaryGUI: e o clasa care realizeaza o interfata grafica, cu ajutorul careia utilizatorul poate sa utilizeze programul fara cunostinte din domeniul programarii, doar prin apasarea unor butoane. Contine urmatoarele elemente de Swing:

- TextField pentru cuvantul pe care vrem sa-l traduca dictionarul sau pentru care vrem sa facem o cautare generica
- Butonul Translate, pe care daca il apasam, in lista de Match vom avea semnificatia cuvantului
- Butonul Search, care daca il apasam, in lista de Match vom avea lista de cuvinte pentru care am avut macthing, dupa cautarea generica in dictionar
- Labeluri, care fac etichetarea componentelor

#### Diagrama UML:



#### Interfata grafica:

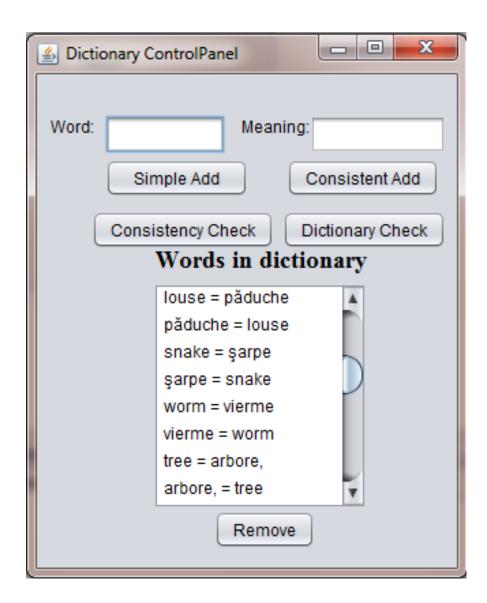
Interfata pentru utilizarea dictionarului, de un utilizator normal, arata astfel:



Aceasta interfata contine urmatoarele elemente:

- TextField pentru cuvantul pe care vrem sa-l traduca dictionarul sau pentru care vrem sa facem o cautare generica
- Butonul Translate, pe care daca il apasam, in lista de Match vom avea semnificatia cuvantului
- Butonul Search, care daca il apasam, in lista de Match vom avea lista de cuvinte pentru care am avut macthing, dupa cautarea generica in dictionar
- Labeluri, care fac etichetarea componentelor

Interfata grafica pentru gestiunea datelor dictionarului, folosit de administratorul aplicatiei arata astfel:



#### Contine urmatoarele elemente:

- TextField pentru cuvantul pe care vrem sa introduce sau pentru care vrem sa face un test de consistenta
- TextField pentru semnificati cuvantului pe care vrem sa introducem sau pentru care vrem sa face un test de consistenta
- Butonul SimpleAdd, pe care daca il apasam realizeaza introducerea cuvantului intre cuvintele continute de dictionar, fara pastrarea proprietatii de consistenta a dictionarului

- Butonul ConsistentAdd, pe care daca il apasam realizeaza introducerea cuvantului intre cuvintele continute de dictionar, cu pastrarea proprietatii de consistenta a dictionarului
- Butonul Remove, pe care daca il apasam va rezulta stergerea cuvantului ales din lista cuvintelor continute de dictionar
- List care contine o lista cu cuvintele si cu semnificatia lor din dictionar
- Labeluri, care fac etichetarea componentelor

Pentru pastrarea corectitudinii continutului dictionarului, am adaugat doua metode de verificare a consistentei, care arata astfel:

```
public boolean checkConsistency(String definedWord, String definingWord)
    if( this.words.containsKey ( definedWord ) )
       if(this.words.containsKey(definingWord) && this.words.get( definedWord
).equalsIgnoreCase( definingWord ) )
         return true;
    return false;
  }
  @Override
  public boolean checkWholeConsistency( )
    String meaning;
    for(String word : this.keys)
       meaning = this.words.get( word );
      if(!this.words.containsKey( meaning ) )
         return false;
     }
    return true;
```

Prima metoda verifica consistenta dictionarului pentru un cuvant introdus in dictionar, iar metoda a doua face o verificare pentru intreg dictionar, verificand fiecare cuvant in parte.

Pentru cautarea cuvintelor dupa un sablon, am folosit REGEX – ul (expresii regulate) in java. Metoda, care realizeaza o cautare corecta arata astfel:

```
public DefaultListModel genericSearch(String pattern) throws PatternSyntaxException
  int i=0;
  int index = 0;
  String tokens[];
  HashMap<String,String> localMap = (HashMap<String,String>) this.words.clone();
  DefaultListModel list = new DefaultListModel();
  pattern = "^" + pattern;
  index = pattern.indexOf("?");
  if(index > = 0)
     pattern = pattern.replace("?", "[a-z]");
  index = pattern.lastIndexOf("]");
  if(index == pattern.length()-1)
     pattern = pattern.concat("$");
  if(!(pattern.endsWith("*")))
     pattern = pattern.concat("$");
  index = pattern.indexOf("*");
  if(index > = 0)
     pattern = pattern.replace("*", "+");
  Pattern r = Pattern.compile(pattern);
  for(String word : this.keys)
     Matcher m = r.matcher(word);
    if (m.find())
       list.addElement(word + " = " + this.words.get(word));
```

```
i++;
}
if(i==0)
list.addElement("NO MATCH!");
return list;
}
```

#### **Utilizarea interfetei:**

Interfata pentru un utilizator normal: dupa pornirea aplicatiei puteti introduce un cuvant in casuta goala, iar daca dati click stanga pe butonul Translate, atunci in fereastra numit Match va aparea semnificatia cuvantului. Daca vreti sa cautati mai multe cuvinte dupa un sablon dat, atunci introduceti sablonul ( sablonul poate sa contina semnul? care inlocuieste un singur caracter sau semnul \* care inlocuieste un sir de caractere oarecare) si dati click stanga pe butonul Search, iar in lista de Match va aparea o lista cu cuvintele care corespund sablonului dat si semnificatia lor.

Interfata pentru administratorul aplicatiei: daca doriti sa adaugati un cuvant nou, introduceti cuvantul si semnificatia acestuia in casutele goale, dupa care apasati butonul Consistent Add, iar dupa apasarea acestui buton, cuvantul va aparea in lista cuvintelor continute de dictionar, iar consistenta dictionarului va fi pastrata. In cazul ni care dati click stanga pe Simple Add, atunci cuvantul va fi introdus in dictionar, dar consistenta dictionarului va fi stricata. Daca doriti sa verificati consistenta dictionarului pentru un cuvant anume, introduceti cuvantul respectiv si dati click pe butonul Consistency Check, dupa care va aparea o fereastra informativa in cazul in care dictionarul e consistent sau o fereastra de eroare in cazul in care dictionarul nu este consistent. Daca doriti sa verificati consistenta intregului dicitonar, atunci dati click stanga pe butonul Dictionary Check, dupa care va aparea o fereastra informativa in cazul in care dictionarul e consistent sau o fereastra de eroare in cazul in care dictionarul nu este consistent. Daca doriti sa stergeti un cuvant din dictionar, atunci alegeti cuvantul de sters in lista cuvintelor si dati click stanga pe butonul Remove, dupa care acest cuvant dispare din lista cuvintelor.

#### 4. Testare

Aceasta aplicatie a fost testata de mai multe ori. Testarea am facut prin introducerea manuala a datelor, verificand si cazurile particulare (de exemplu, daca am vrut sa introduc cuvinte deja existente in dictionar, atunci introducerea a esuat, dupa introducerea unui cuvant fara pastrarea consistentei, Dictionary Check mi-a dat eroare, am folosit sabloana pentru cautare si intotdeauna mi-a dat o lista de cuvinte care respecta sablonul dat etc. ) iar rezultatul testarii a fost intotdeauna conform realitatii, fara greseli.

Pentru o testare mai buna am facut aple si la alte persoane, care au incercat utilizarea aplicatiei cautand greseli, iar daca au aparut ceva greseli pe care pana atunci nu am luat in considerare, pe care pana atunci nu am gasit, am reformulat codul pentru corectarea acestor greseli. In prezent aplicatia merge fara greseli.

#### 5. Rezultate

După testarea programului vedem că programul functionează perfect, verifică datele de intrare , afișează rezultatele pe interfața grafică, si efetuează comenzile primite de la utilizatorul programului.

#### 6. Concluzii, ce s-a invatat din tema, dezvoltari ulterioare

Lucrand la acest proiect am vazut avantajele folosirii preconditiilor si postconditiilor, care ne ajuta in creearea unei aplicatii cat mai corecte din punct de vedere a functionalitatii precum si din punct de vedere a logicii si a cerintelor de Bussiness si Management. Am invatat sa creez exceptii proprii, si am inteles mai bine folosirea exceptiilor si avantajele lor. Pe langa acestea am mai invatat folosirea expresiilor regulate, folosirea Regex-ului in java si folosirea Map-urilor in java, mai ales HashMap – ul , pentru ca in proiectul meu am folosit acest tip de Map.

### 7. Bibiliografie

#### Carti:

Thinking in java – fourth edition Bruce Eckel President, MindView, Inc.

Effective Java™

Second Edition

#### Internet:

https://docs.oracle.com/javase/tutorial/collections/interfaces/map.html

https://www.youtube.com/ - turoriale

http://stackoverflow.com/

http://docs.oracle.com/javase/tutorial/