**Muntean Andrei 30227**

**DICTIONAR**

**DE SINONIME**

1. **Obiectivul temei**

*Cerinţa:*

Să se implementeze un dicţionar explicativ sau de sinonime al limbii române sau engleze. Pentru implementare sa va folosi Java Collection Framework Map. Definiţi şi implementaţi o interfaţă specifică (populare, adăugare, salvare, ştergere, copiere etc.) împreună cu utilitare pentru procesarea dicţionarului. Consideraţi următoarele utilitare:

* Verificarea consistenţei dicţionarului. Un dicţionar este consistent dacă toate cuvintele folosite în definirea unui cuvânt sunt la rândul lor definite.
* Căutarea să se poată facă şi utilizând \*(cu semnificaţia orice string, chiar şi null) şi ?(cu semnificaţia un caracter).

Notă: se vor folosi contracte, invarianţi, assert-uri, javadoc etc.

Obiectivul este unul destul de bine stabilit încă din cerinţa temei. Se doreşte implementarea unui dicţionar care să vină totodată cu o serie de facilităţi prin care să se poată manipula acesta. A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A

**2. Analiza problemei**

Problema, privită complex poate fi extrem de dificilă cand vine vorba de rezolvarea anumitor cerinţe precum consistenţa dicţionarului. De aceea, ceea ce-mi propun eu este să fac doar anumite operaţii de bază asupra dicţionarului, iar când vine vorba de ceva mai complex să rezolv doar o parte din cerinţă.

Pentru implementare pare foarte potrivita o interfata grafica care să ajute utilizatorul în folosirea aplicaţiei.

Ca şi constrângeri acestea vor fi destul de mari la număr. De exemplu, de la bun început fişierul din care se populează dicţionarul este unul de dimensiune redusă. Asta se datorează în principal faptului că, pentru a putea verifica consistenţa, am fost nevoit să fac un compromis, atât în dimensiunea fişierului cât şi în rigurozitatea cu care am aplicat algoritmul de verificare a consistenţei. Mai mult, cuvintele conţinute în fişier sunt mult truncate pentru că am încercat să obţin un fişier care este deja consistent, pentru a vedea că algoritmul de verificare a consistenţei funcţionează. Altfel, acest lucru ar fi fost aproape imposibil.

În ceea ce priveşte construirea claselor, am ales să fac doua la număr, fiecare urmărind să resolve un anumit aspect legat de aplicaţie, astfel: o clasă să se ocupe de partea grafică (clasa Interfata, iar cealalta clasă să implementeze anumite funcţii pentru dicţionar. Ca şi operaţii am ales să implementez:

* Căutarea normală sau după un pattern;
* Adăugarea, respectiv ştergerea unui cuvânt din dicţionar;
* Slavarea unui cuvânt în fişier pentru a putea fi folosit în dicţionar şi la o eventuală rulare viitoare;
* Verificarea consistenţei dicţionarului;
* Popularea dicţionarului din fişier.

**3. Proiectare (diagrame UML, proiectare clase, algoritmi, interfaţa utilizator)**

*Clasa InterfataUtilizator*

După cum sugerează şi numele, această clasă se ocupă interfaţa grafică a programului. Aceasta conţine, pe lângă declaraţiile şi iniţializările componentelor grafice şi declararea unor clase interne ce se ocupă de rezolvarea acţiunilor pe butoane.

*Clasa Dictionar*

Este cea mai bogată clasă, furnizând marea majoritate a metodelor pentru rezolvarea anumitor task-uri asupra dicţionarului, precum: adăugarea unui nou cuvânt, ştergerea unui cuvânt deja existent în dicţionar, căutare, verificare consistenţă etc. De asemnea, conţine anumite metode ce se ocupă de verificarea calităţii datelor, metode ce le ajută pe celelalte în rezolvarea task-urilor. De asemenea aceasta clasa implementeaza interfatza InterfDictionar. In care vor fi scrise metodele publice ale clasei Dictionar.

*Continutul fisierului de populare*

om=persoana fiinta

fiinta=animal viata om

viata=trai

trai=existenta

animal=creatura dobitoc faptura fiinta

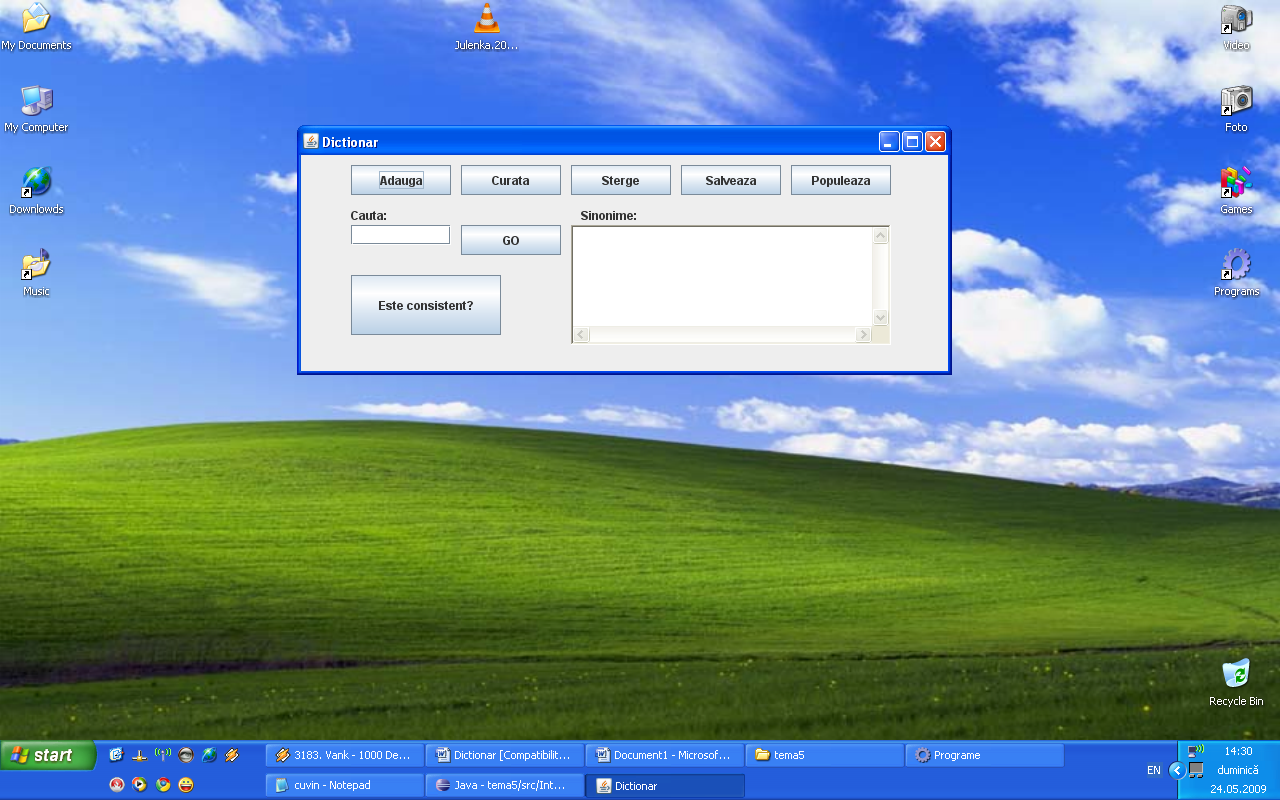
creatura=individ om

dobitoc=animal

individ=persoana

persoana=om figura fata

*Interfata:*

**

Interfaţa este una grafică şi poate fi observată în figura de mai sus. Fiecare buton corepunde unei anumite utilităţi. S-a folosit pentru căuate un jTextField, iar pentru afişare şi alte introduceri de date un TextArea.

Pentru căutare, normal sau special, se va scrie un şir de caractere sub “Cauta:” şi se va apăsa butonul de GO. Pentru adăugare, se va scrie sub “Cauta”cuvantul si sinonimele sub “Sinonime” şi se va apăsa butonul Adauga. Pentru ştergere, se va scrie cuvântul şi se va apăsa Sterge. Pentru salvare se va apăsa Salveaza, pentru verificarea consistenţei Este consistent?, iar pentru popularea din fişier Populeaza. Curata şterge textul aflat în TextArea.

A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A

A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A

**4.Implementarea claselor**

*Clasa InterfUtilizator*

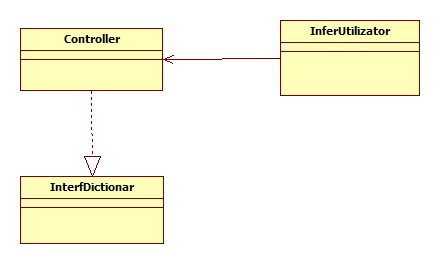
Aceasta clasa extinde clasa JFrame şi se ocupa cu implementarea interfeţei grafice a programului. Este totodată şi clasa care contine metoda main si cea în care se cheamă utilitarele pentru rezolvarea anumitor cerinţe asupra dicţionarului.Componentele grafice sunt declarate ca şi variabile globale nestatice, care mai apoi vor fi instanţiate în cadrul unei metode. Pe lângă acestea sunt declarate global un obiect de tip Dictionar folosit pentru chemarea nestatică a metodelor din aceste clase. Contructorul implicit este suprascris: în el se setează Layout-ul, titlul, close operation şi dimensiunile ferestrei.

Am implementarea metodei Actionperformed idn interfata ActionListner in care apelez metodele din clasa Dictionar asupra obiectului de tip dictionar care il am instantziat in clasa in dependentza de ce buton a fost apasat. Si alte metode in care formez diferite mesaje pentru utilizator esle sunt statice astfel nu trebuie instantziat obiect pentru a le apela si metode private clasei de returnare sau adaugare la zonele de text a unor siruri de caractere sau din ele.

*Clasa Dictionar*

Clasa Utilitare se ocupă de implementarea unor funcţii specifice dicţionarului, funcţii precum adăugarea şi ştergerea unui cuvânt din dicţionar, căutarea simplă sau după un pattern, verificarea consistenţei dicţionarului etc. şi alte metode ce ajută la verificarea corectitudinii operaţiilor efectuate asupra dicţionarului.

**Diagrama UML**

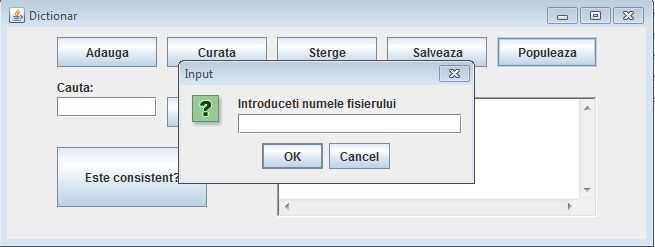


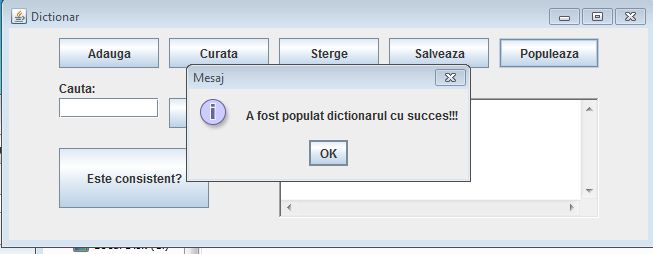
**5. Testarea**

Dupa lansarea programului vedem urmatoare fereastra :

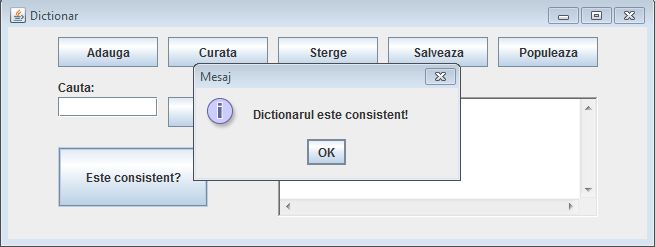


Pentru a lucra cu dictionarul el trebuie sa fie populat dintr-un file TXT apasand butonul „Populeaza”.

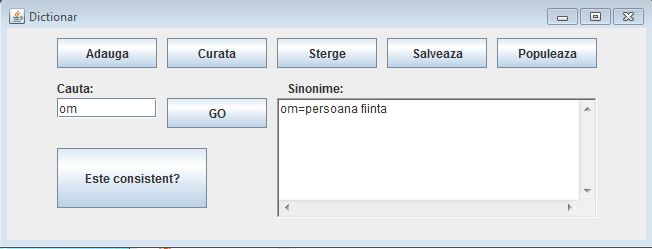




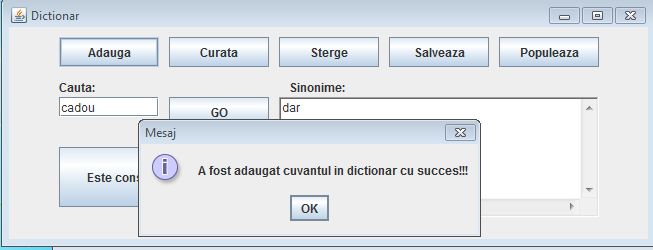
Pentru a verifica consistenta dictionarului putem apasa butonul „Este consistent?” si programul ne va arata un mesaj afirmativ sau negativ :



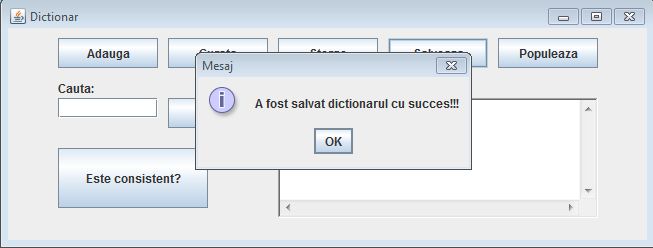
Ca sa facem cautarea in dictionar trebuie sa introducem cuvantul sau prima parte a cuvantului in forma „Cauta” astfel :



Pentru a adauga un cuvant,se introduce cuvantul dorit in forma „Cauta” si sinonimul in forma „Sinonime” dupa ce se apasa butonul „Adauga”:



Pentru a salva modificarile trebuie sa apasam butonul „Salveaza” :



**6.Dezvoltări ulterioare**

Dezvoltări ulterioare, ca mai întotdeauna, pot fi nenumărate. Unele părţi la care se poate dezvolta pornind de la proiectul data ar putea fi:

O verificare mult mai complex a consistenţei dicţionarului (mai multe criterii);

Creşterea dimensiunii şi a calităţii informaţiei din fişierul din care se va face popularea. De exemplu, se poate adăuga pentru fiecare cuvânt ce parte de vorbire este;

Implementarea unei grafici mai atrăgătoare pentru utilizator;

Creşterea complexităţii datelor, cum ar fi adăugarea de sinonime, antonime etc. pentru un cuvânt dat.

Posibilitatea de folosire a mai multor fişiere pentru populare;

Etc.