

Trabalho 2: Corrector ortográfico



Trabalho elaborado por:

Marcelo Babau nº 30372

Docentes:

Ligia Ferreira

Introdução

Este trabalho tem o objetivo final de implementar um corretor ortográfico, para isso vamos manipular estruturas de dados como as Tabelas de Hash. Para o desenvolvimento do corrector ortografico é dado um ficheiro .txt com o dicionario, uma palavra por linha. Ao correr o programa é suposto criar um ficheiro .txt com as palavras mal escritas e as respectivas sugestões. Sugestões criadas através da inserção, remoção e troca de letras.

Desenvolvimento

O trabalho tem as seguintes classes:

SpellChecker.java

Hash_Linear.java

HashDouble.java

HashQuadratica.java

HashTable.java

HashElemento.java

Main.java

InvalidNodeExcepton.java

Linkedlist.java

LinkedListIterator.java

Node.java

Palavra.java

Foi fornecido o ficheiro com o dicionario o ficheiro txt, ao qual passamos como um dos argumentos no construtor : **public SpellChecker(String dictFile, String errorFile, String docFile)** onde **dictFile** é o nome do ficheiro do Dicionário, **errorFile** é o nome do ficheiro que se vai criar com a palavra incorreta e respectivas sugestões e o **docFile** que é o nome do ficheiro que queremos corrigir.

A classe SpellChecker, **Sc**, é a que trata todas as operações como criar o dicionário com o método

HashQuadratica<String> hqDic = Sc.getHashTable() que cria a respectiva tabela de hash com o dicionário, neste caso HashQuadratica.

Nesta mesma classe temos o metodo **Sc.spellCheck("fileAcorrigir.txt")** que abre o ficheiro que contém o texto e adiciona a uma lista ligada todas as palavras incorretas. Em seguida é chamado o **criarSugestoes()** que cria uma lista ligada com sugestões, **palavraSug()** este metodo chama outros metodos o **sugInserere**, **sugRemove** e **sugTroca** que trata das operações de inserção, remoção e troca de caracteres sobre cada palavra incorreta.

Depois de encontrar as sugestões, é criada uma instancia da classe Palavra, **Palavra p = new Palavra(String palavraIncorreta, LinkedList Sugestoes)** sendo adicionada essa **Palavra** à tabela de **Hash_Linear<Palavra>**, que vai conter todas as palavras incorretas e sugestões para essas palavras.

Por fim temos o metodo **geraFicheiro(HashTable<Palavra> sugestoes, String errorFile)** que recebe a tabela hash de Palavras e uma string errorFile, que vai ser o nome do ficheiro que vamos criar com as **Palavras** presentes na tabela de hash sugestões.

Conclusão

Com este projecto, de desenvolver um corrector ortográfico, os meus conhecimentos, sobre manipulação de tabelas de dispersão e sobre os 3 tipos que existem, evoluíram bastante.

Alguns bugs foram encontrados, ex: palavras como "não" é vista como incorretas por causa da pontuação, e por alguma razão a ultima palavra de cada linha é lida como incorreta, talvez devido a algum "\n" no fim de cada linha... Também mete null no inicio das sugestões pois estas estão guardadas numa listaLigada sendo o null referente ao header da lista.

Todo o trabalho foi implementado de raiz, com as listas ligadas já implementadas do trabalho anterior.

Fiquei a compreender melhor as tabelas de dispersão e a dominar melhor a linguagem JAVA e também as operações de escrita e leitura de ficheiros.

