Dictionary システム
Version 1.0

平成24年12月10日

田中　隆彦

本書は、Java 言語で実装された単語検索、単語比較ツールである Dictionary システムについて説明するものです。このため、本書では、Java 言語の知識、及びコンピュータシステムについての一般的な知識を持っていることを前提としています。

本書は、下記オペレーティングシステム上での動作を保証するもので、その他のバージョンでの動作を保証するものではありません。

Microsoft Windows XP 5.1.26
Mac OS

未対応バージョンへの対応、問題の問い合わせ等は下記メールアドレスにメールで御連絡ください。

uewxiun@pb3.so-net.ne.jp

平成24年12月10日

田中　隆彦
uewxiun@pb3.so-net.ne.jp

Java と Oracle は、oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。Microsoft と Windows XP は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。UNIX は、The Open Group の米国における登録商標です。Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

1.　システムの概要

文字検索方法として、高速な二分岐検索法が一般的に使用されています。その検索のための二分岐を構築する方法は特に高速な方法が今まで一般的には存在しなかった。このため、限られた語句を使用する辞書で、動的な追加、削除を可能とするものを通常のソフトウェアシステムが内包することは難しいものとなっていました。本システムは、新たに考案されたに分岐構築、改変のための高速アルゴリズムを使用して、動的な追加、削除が簡単に出来る簡易仮名漢字変換ライブラリとして構築されたものです。

本システムは、簡易仮名漢字変換システムとして、高速な単語検索・登録・削除のための Java 言語のライブラリとユーザインタフェースを提供するものです。ライブラリとして使用する際の辞書は、附随の GUI コマンドを使用して対話的に、データを追加して御使用頂けます。通常のアプリケーション内部で、部分的な日本語の読みによる並べ替えのためだけに、必要な単語だけを登録して、御使用頂けるフレキシブルなライブラリシステムです。

本システムのユーザインタフェースは、Java の AWT(Abstract Window Toolkit)クラスを使用して実装されているため、UNIX/Linux でも、Windows でも一般のコマンドと同様に御使用頂けます。

2.　ユーザインタフェース

本システムは、ユーザインタフェースとしては、ウインドウベースのインタフェースとコマンドラインベースのインタフェースの2種類を提供しています。

Java AWT(Abstract Window Toolkit)ベースのユーザインタフェースの起動は、次のようにして実行出来ます。

```
# java -classpath CLASSPATH dict.window.DictionaryCommand [-g|[-l eng |-l jpn]]
```

-g　オプションはデバッグメッセージン出力を要求するオプションです。
-l　オプションは、辞書のメッセージの言語を指定します。
　現在は、英語と日本語の二種類だけをサポートしています。
　日本語の指定方法は、-l jpn です。-l eng を指定すれば英語を
　指定した事になります。

ユーザインタフェースとして、コマンドユーザインタフェースも兼ね備えているため、データの初期化等の冗長になり易い作業を、スクリプト等で効率よく処理する事も可能です。コマンドインタフェースの起動方法は次のようなものです。

```
# 追加
# java -classpath CLASSPATH dict.JapaneseDictionary a dictionary word pronounce

# 削除
# java -classpath CLASSPATH dict.JapaneseDictionary d dictionary word pronounce

# 読みの参照
# java -classpath CLASSPATH dict.JapaneseDictionary g dictionary word

# 単語の参照
# java -classpath CLASSPATH dict.JapaneseDictionary c dictionary pronounce
```

3.　クラスのリファレンス

JapaneseDictionary クラス
日本語の漢字の単語とその読みを管理するためのクラスです。内部の辞書を利用して、漢字の単語同士をその読みを使用して「あいうえお」順で比べることができるものです。

1つの漢字の単語に対して複数の読みは登録出来ません。
しかし、1つの読みに対して複数の漢字の単語は登録可能です。

public static int compareKanji(String s1, String s2) throws Exception

二つの漢字を含む単語をその登録されている読みで「あいうえお」順の比較するためのメソッドです。それぞれ読みが登録されている必要があります。
s1の文字が「あいうえお」の順番で早い文字の場合には負の値、同じ文字列で、s1の長さが短い場合には、-1、長い場合には1を返します。

public String[] getCandidate(String entry) throws Exception

単語の読みに対して、漢字の単語のリストを返すメソッドです。

public String findEntry(String entry) throws Exception

漢字を含む単語に対して、読みを返すメソッドです。

public String getKey(int order) throws Exception

読みの並び順の番号に対して、その漢字を含む単語を返します。

public String getValue(int order) throws Exception

読みの並び順の番号に対して、その読みを返すメソッドです。

public void addEntry(String word, String pronounce) throws Exception

漢字を含む単語とその読みのエントリを追加します。

public void deleteEntry(String word, String pronounce) throws Exception

漢字を含む単語とその読みのエントリを削除します。

public String[] getEntry(int position) throws Exception

漢字を含む単語とその読みのエントリを返します。

StringMatch クラス
文字列を順番に並べて管理するクラスです。平仮名、濁音、半濁音、促音、とカタカナは「あいうえお」順では同一と見なし、同じ場合は、 次の順番で並べ替えられます。

1. 清音平仮名
2. 清音片仮名
3. 濁音平仮名
4. 濁音片仮名、
5. 半濁音平仮名
6. 半濁音片仮名
7. 促音平仮名
8. 促音片仮名

public static int compareJapanese(String s1, String s2) throws Exception

単語を比較するメソッドです。

public void add(String sentence) throws Exception

単語を登録します。

public int delete(String key) throws Exception

指定された単語の削除します。

public int delete(int position) throws Exception

登録順番で指定された単語の削除します。

public String getMessage(int position) throws Exception

登録順番で指定された位置の単語を返します。

public String getKey(int order) throws Exception

　並び順で指定された位置の単語を返します。

public String getItem(int index) throws Exception

　検索順で指定された位置の単語を返します。

public int getPositionInKey(int order) throws Exception

　指定された並び順の位置の単語の登録順の番号を返します。

public int[] getOrder(String key) throws Exception

　指定された単語の並び順での位置のリストを返します。

public int[] getIndex(String key) throws Exception

　指定された単語の検索順での位置のリストを返します。

public int[] getPosition(String key) throws Exception

　指定された単語の登録順での位置のリストを返します。

public int getCount()

　登録単語数を返します。

public int getCount(String key) throws Exception

　指定された単語と同じ登録単語の数を返します。

public void loadMessages(ObjectInputStream object_input_stream) throws Exception

指定された ObjectInputStream を通して、インデックス配列と登録単語配列をそれぞれのサイズの Integer のオブジェクトと共に格納されているものとして辞書データを読み込みます。

public void writeMessages(ObjectOutputStream object_output_stream) throws Exception

指定された ObjectOutputStream を通して、インデックス配列と登録単語配列をそれぞれのサイズの Integer のオブジェクトと共に書き込みます。

public void printOrder()

並び順に登録単語を表示します。