第5回演習課題

以下の課題をこなし，提出要領に従って提出しなさい．

問1 （基礎）：

以下のプログラムは，Javaのコーディング規約に従っていない．コーディング規約に従ってコードの修正(リファクタリング）を行いなさい．

※注意： 名前を変更する場合は，Eclipse上でその名前をドラッグして「右クリック」→「リファクタリング」→「名前の変更」を使うこと．単純にキーボードから書き換えると，書き換え忘れが頻繁に起こり，悲惨なことに．．．

kadai3_1.java

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashMap;

/*
 * 第3回演習問１：英和辞書プログラム．
 */
public class kadai3_1 {
/*辞書データ．<単語,日本語訳>のマップ */
HashMap <String, String> m = new HashMap <String,String>();

/*
 * 辞書ファイルをロードして，マップに読み込む
 */
public void Dictionary_File(String f) {
m.clear();//一旦マップを空にする

try {
//ファイルをオープン
BufferedReader br = new BufferedReader(new FileReader(f));
String line;

//1行ずつ読み出して
while ((line = br.readLine())!=null) {
//タブで分割
String [] data = line.split("\t");
if (data.length >= 2) {
```

```
//マップに登録する
m.put(data[0], data[1]);
}
}
br.close();
} catch (FileNotFoundException e) {
System.err.println("辞書ファイル " + f + "が見つかりません．");
System.err.println("終了します．");
System.exit(1);
} catch (IOException e) {
System.err.println("辞書ファイル読み込み中，IO例外が発生しました．");
System.err.println("終了します．");
System.exit(1);
}


}

/*
 * キーボードから文字列を受付け，入力された文字列を返す．
 * 例外が発生すればやり直す．
 */
public String mojiretsu() {
String str;
try {
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
str = br.readLine();
} catch (IOException e) {
System.err.println("IO例外です．再入力．");
str = mojiretsu();
}
return str;
}

/*
 * 単語を検索する
 */
public void searchBooks() {
System.out.println("■簡易英和辞典：");
while (true) {
System.out.println();
System.out.print("◎単語入力？(空行で終了)：");
String word = mojiretsu();

//空行のチェック
if (word.equals("")) {
System.out.println("  #終了します．");
break;
}

//検索
if (m.containsKey(word)) {
```

```
String meaning = m.get(word);
System.out.println(word);
System.out.println("  " + meaning);
} else {
System.out.println("  そのような単語はありません．");
}
}

}

/*
 * メイン関数
 */
public static void main(String[] args) {
kadai3_1 k = new kadai3_1();
k.Dictionary_File("eiwa.txt");
k.searchBooks();
}

}
```

問2 (応用)：

第5回・例題2のクラス図に従って，友達管理アプリケーション（拡張版）を実装しなさい．ファイルへのリストの保存(save)，読み出し(load)は余裕がある人だけ取り組むこと．

- ユーザは，メニューから以下の操作のうち1つを選択し，プログラムを利用する．終了が選択されるまで，繰り返し行う．
- [1: 友達リストを見る] 現在登録されている友達リストを表示する．
- [2: 友達を追加する] 名前，電話番号，メールアドレスを入力して，友達をリストに追加する．
- [3: 友達を削除する] 番号を入力して，リストから友達を削除する．
- [4: 友達の詳細を見る] 番号を入力して，友達の詳細情報を確認する．
- [0: 終了する] プログラムを終了する．

```
＃実行例


■友達リストを管理します．コマンドを入れてください．
[1: 友達リストを見る，2: 友達を追加する，3:友達を削除する，4:友達の詳細を見る，0:終了する]
```

```
1
[リスト] 友達リストを表示します.
---------------------------------
  no:  name
---------------------------------
---------------------------------

■友達リストを管理します. コマンドを入れてください.
[1: 友達リストを見る, 2: 友達を追加する, 3:友達を削除する, 4:友達の詳細を見る, 0:終了する]
2
[追加] 友達を追加します.
   名前：Masahide Nakamura
   電話番号：078-123-4567
   メールアドレス：masa-n@cs.kobe-u.ac.jp
Masahide Nakamuraさんを追加しました.

■友達リストを管理します. コマンドを入れてください.
[1: 友達リストを見る, 2: 友達を追加する, 3:友達を削除する, 4:友達の詳細を見る, 0:終了する]
2
[追加] 友達を追加します.
   名前：Haruhisa Maeda
   電話番号：090-1111-2222
   メールアドレス：maeda@aniki.org
Haruhisa Maedaさんを追加しました.

■友達リストを管理します. コマンドを入れてください.
[1: 友達リストを見る, 2: 友達を追加する, 3:友達を削除する, 4:友達の詳細を見る, 0:終了する]
2
[追加] 友達を追加します.
   名前：Shota Nakatani
   電話番号：0786-98-7654
   メールアドレス：shota-n@example.com
Shota Nakataniさんを追加しました.

■友達リストを管理します. コマンドを入れてください.
[1: 友達リストを見る, 2: 友達を追加する, 3:友達を削除する, 4:友達の詳細を見る, 0:終了する]
1
[リスト] 友達リストを表示します.
---------------------------------
  no:  name
---------------------------------
   0:  Masahide Nakamura
   1:  Haruhisa Maeda
   2:  Shota Nakatani
---------------------------------

■友達リストを管理します. コマンドを入れてください.
[1: 友達リストを見る, 2: 友達を追加する, 3:友達を削除する, 4:友達の詳細を見る, 0:終了する]
4
```

```
[詳細] 何番の友達を確認しますか？1
---------------------------------
  名　前：  Haruhisa Maeda
  電　話：  090-1111-2222
  メール：  maeda@aniki.org
---------------------------------

■友達リストを管理します．コマンドを入れてください．
[1: 友達リストを見る，2: 友達を追加する，3:友達を削除する，4:友達の詳細を見る，0:終了する]
3
[削除] 何番の友達を削除しますか？0

■友達リストを管理します．コマンドを入れてください．
[1: 友達リストを見る，2: 友達を追加する，3:友達を削除する，4:友達の詳細を見る，0:終了する]
1
[リスト] 友達リストを表示します．
---------------------------------
  no:  name
---------------------------------
   0:  Haruhisa Maeda
   1:  Shota Nakatani
---------------------------------

■友達リストを管理します．コマンドを入れてください．
[1: 友達リストを見る，2: 友達を追加する，3:友達を削除する，4:友達の詳細を見る，0:終了する]
0
終了します．
```

ヒント：

- 第3回の演習のコードを無理やり改造して作らないこと．1から作った方がすっきりする．
- クラス図から先にコードのスケルトン（骨組み）を作ってしまおう．メソッドの中身は後回し．
- Friendは友達データ，FriendListは友達データベース，FriendManagerはユーザとのインタフェースというように責務を分けている．
- KeyBoardクラスは汎用的なユーテリティ（道具）クラス．staticメソッドは，int num = KeyBoard.inputNumber()のように呼び出す．

問2（発展）：　（余裕のある人だけ取り組みなさい）

複数のリストを切り替えて管理できるようにするには，どうすればよいだろうか？（例：仕事用リスト，プライベート用リスト，SNS用リスト） 必要ならクラ

ス設計を見直して，アプリを拡張しなさい.

masa-n@cs.kobe-u.ac.jp

Last updated: 05/25/2020 13:20:19