import edu.duke.\*;

import java.util.\*;

public class GladLib {

private ArrayList<String> adjectiveList;

private ArrayList<String> nounList;

private ArrayList<String> colorList;

private ArrayList<String> countryList;

private ArrayList<String> nameList;

private ArrayList<String> animalList;

private ArrayList<String> timeList;

private ArrayList<String> wordsReplaced;

private ArrayList<Integer> count;

private ArrayList<String> fruitList;

private ArrayList<String> verbList;

private Random myRandom;

private static String dataSourceURL = "http://dukelearntoprogram.com/course3/data";

private static String dataSourceDirectory = "data";

public GladLib(){

initializeFromSource(dataSourceDirectory);

wordsReplaced = new ArrayList<String>();

count = new ArrayList<Integer>();

myRandom = new Random();

}

public GladLib(String source){

initializeFromSource(source);

wordsReplaced = new ArrayList<String>();

count = new ArrayList<Integer>();

myRandom = new Random();

}

private void initializeFromSource(String source) {

adjectiveList= readIt(source+"/adjective.txt");

nounList = readIt(source+"/noun.txt");

colorList = readIt(source+"/color.txt");

countryList = readIt(source+"/country.txt");

nameList = readIt(source+"/name.txt");

animalList = readIt(source+"/animal.txt");

timeList = readIt(source+"/timeframe.txt");

fruitList = readIt(source+"/fruit.txt");

verbList = readIt(source+"/verb.txt");

}

private String randomFrom(ArrayList<String> source){

int index = myRandom.nextInt(source.size());

return source.get(index);

}

private String getSubstitute(String label) {

if (label.equals("country")) {

return randomFrom(countryList);

}

if (label.equals("color")){

return randomFrom(colorList);

}

if (label.equals("noun")){

return randomFrom(nounList);

}

if (label.equals("name")){

return randomFrom(nameList);

}

if (label.equals("adjective")){

return randomFrom(adjectiveList);

}

if (label.equals("animal")){

return randomFrom(animalList);

}

if (label.equals("timeframe")){

return randomFrom(timeList);

}

if (label.equals("fruit")){

return randomFrom(fruitList);

}

if (label.equals("verb")){

return randomFrom(verbList);

}

if (label.equals("number")){

return ""+myRandom.nextInt(50)+5;

}

return "\*\*UNKNOWN\*\*";

}

private String processWord(String w){

int first = w.indexOf("<");

int last = w.indexOf(">",first);

int index = wordsReplaced.indexOf(w);

int feq = 1;

count.set(index, feq + 1);

if ((first == -1 || last == -1 ) && (index == -1 )) {

return w;

}

String prefix = w.substring(0,first);

String suffix = w.substring(last+1);

String sub = getSubstitute(w.substring(first+1,last));

//wordsReplaced.add(sub);

return prefix+sub+suffix;

}

private void printOut(String s, int lineWidth){

int charsWritten = 0;

for(String w : s.split("\\s+")){

if (charsWritten + w.length() > lineWidth){

System.out.println();

charsWritten = 0;

}

System.out.print(w+" ");

charsWritten += w.length() + 1;

}

}

private String fromTemplate(String source){

String story = "";

if (source.startsWith("http")) {

URLResource resource = new URLResource(source);

for(String word : resource.words()){

story = story + processWord(word) + " ";

}

}

else {

FileResource resource = new FileResource(source);

for(String word : resource.words()){

story = story + processWord(word) + " ";

}

}

return story;

}

private ArrayList<String> readIt(String source){

ArrayList<String> list = new ArrayList<String>();

if (source.startsWith("http")) {

URLResource resource = new URLResource(source);

for(String line : resource.lines()){

list.add(line);

}

}

else {

FileResource resource = new FileResource(source);

for(String line : resource.lines()){

list.add(line);

}

}

return list;

}

public void findNoOfWordsReplaced(){

wordsReplaced.clear();

count.clear();

//for (int k = 0; k < wordsReplaced.size(); k++);

//System.out.println(NoOfWordReplaced.size());

int count = count;

System.out.println("Total no of words replaced : " + count);

}

public void makeStory(){

System.out.println("\n");

String story = fromTemplate("data/madtemplate2.txt");

printOut(story, 60);

System.out.println("\n");

findNoOfWordsReplaced();

}

}