|  |
| --- |
| **2019년 11월 18일 실습보고서** |
| **5조 이름 : 송재원 조윤직 김영준 안정근** |
| **실습자료1 : [Wordtest]** |
| **소스코드** |
| //Wordtest.java  **package** konkuk;  **import** java.io.\*;  **import** java.util.\*;  **public** **class** Wordtest {  **static** Random *r* = **new** Random();  **static** Scanner *sc* = **new** Scanner(System.***in***);  **protected** **static** LinkedHashMap<String, Word> *words*=**new** LinkedHashMap<>();  **public** **static** **void** fileopen() **throws** FileNotFoundException {  File file = **new** File("words.txt");  Scanner scans = **new** Scanner(file);    **while** (scans.hasNextLine()) {  String eng = scans.nextLine().trim();  String[] mean = (scans.nextLine().trim()).split("/");  **for** (**int** i = 0; i < mean.length; i++) {  mean[i] = mean[i].trim();  }  **if** (!*words*.containsKey(eng)) {  *words*.put(eng, **new** Word(eng, mean));  }  }  scans.close();  }  // 인덱스로 접근 할수 없는 Linkedhashmap에 iterator로 n번째 밸류에 접근을 시켜주는 함수  **public** **static** Word getWords(**int** num) {  Set<String> set = *words*.keySet();  Iterator<String> it = set.iterator();  **int** count = 0;  **while** (it.hasNext()) {  **if** (count == num) {  **return** *words*.get(it.next());  } **else** {  it.next();  count++;  }  }  **return** **null**;  }  **public** **static** **void** AddWords() {  String addEng;  System.***out***.print("추가할 영단어를 입력하세요 : ");  addEng = *sc*.nextLine().trim();  **if** (!(*words*.containsKey(addEng))) {  System.***out***.print("단어의 뜻을 입력하세요(뜻이 여러개일 경우 '/'로 구분해주세요) :");  String[] addMean = (*sc*.nextLine().trim()).split("/");  *words*.put(addEng, **new** Word(addEng, addMean));  **for** (String m : addMean) {  System.***out***.println(addEng + " : " + m);  }  } **else** {  System.***out***.println("등록되어 있는 단어입니다.");  }  }  **public** **static** **void** main(String[] args) **throws** FileNotFoundException {  // **TODO** Auto-generated method stub  *fileopen*();  **int** num;  System.***out***.println("5조 송재원 조윤직 김영준 안정근 입니다.");  **while** (**true**) {  System.***out***.println(*words*.size());  System.***out***.println("1) 주관식 2) 객관식 3) 단어 추가 4) 퀴즈 빈출 단어 5) 종료");  num = *sc*.nextInt();  *sc*.nextLine();  **if** (num == 1) {  *Subjective*();  } **else** **if** (num == 2) {  *ObjectiveCorrect*();  } **else** **if** (num == 3) {  *AddWords*();  }**else** **if**(num==4) {  *Quiz\_words*();    } **else** **if** (num == 5) {  System.***out***.println("종료합니다.");  **break**;  } **else** {  System.***out***.println("다시 입력해 주세요");  }    }  }  **public** **static** **void** Quiz\_words() {  System.***out***.print("오름차순으로 할까요 (true/false)입력 : ");  **boolean** dd=*sc*.nextBoolean();  *sc*.nextLine();  System.***out***.print("몇 개를 출력할까요? : ");  **int** count =*sc*.nextInt();  List<Word> list=**new** ArrayList<Word>();    list.addAll(*words*.values());  **if**(dd)  Collections.*sort*(list);  **else**  Collections.*sort*(list,Collections.*reverseOrder*());    **for**(**int** i=0;i<count;i++) {  **if**(!list.get(i).zeroCount())  System.***out***.println(list.get(i));  }    }  **public** **static** **void** ObjectiveCorrect() {  **int** w=0;  **boolean**[] delWord = **new** **boolean**[*words*.size()];  **int** repeat=0;  **for**(**int** j=0;j<5;j++) {  **int** num;  **do** {  **if**(repeat==*words*.size()) {  System.***out***.println("단어장의 모든 단어가 사용되었습니다.");  **return**;  }  num = *r*.nextInt(*words*.size());  }**while**(delWord[num]);  repeat++;  delWord[num]=**true**;    Word[] testWord = **new** Word[5];  testWord[0] = *getWords*(num);  testWord[0].upCount();  Word correctWord = testWord[0];  **boolean**[] useWord = **new** **boolean**[*words*.size()];  useWord[num]=**true**;  **for** (**int** i=1;i<testWord.length;i++) {  **int** rand = *r*.nextInt(*words*.size());  **if**(useWord[rand]) {  i--;  **continue**;  }**else** {  testWord[i]=*getWords*(rand);  useWord[rand]=**true**;  }  }  **for**(**int** i=0;i<100;i++) {  Word tempWord;  **int**[] mixNum = {*r*.nextInt(4),*r*.nextInt(4)};  **if**(mixNum[0]==mixNum[1]) {  i--;  **continue**;  }**else** {  tempWord=testWord[mixNum[0]];  testWord[mixNum[0]]=testWord[mixNum[1]];  testWord[mixNum[1]]=tempWord;  }  }  System.***out***.println();  System.***out***.println("\*객관식 문제\*");  System.***out***.println((j+1)+"번 :["+correctWord.getWord\_e()+"]");  **for**(**int** i=0;i<testWord.length;i++) {  System.***out***.print("- "+(i+1)+"번 : ");  **for**(**int** k=0;k<testWord[i].getMean().length;k++) {  System.***out***.print(testWord[i].checkMean(k));  **if**(k<testWord[i].getMean().length-1) {  System.***out***.print(" / ");  }  }  System.***out***.println();  }  System.***out***.print("-단어뜻 : ");  **int** userNum;  **while**(**true**) {  userNum = *sc*.nextInt();  **if**(userNum<1||userNum>5) {  System.***out***.println("1~5번까지의 보기중에서 선택해주세요.");  **continue**;  }**else** {  **break**;  }  }    userNum-=1;  **boolean** check=**true**;  **for**(**int** i=0;i<testWord[userNum].getMean().length&&i<correctWord.getMean().length;i++) {  **if**(testWord[userNum].checkMean(i).equals(correctWord.checkMean(i))) {  System.***out***.println("정답입니다.");  w+=1;  check=**false**;  **break**;  }  }  **if**(check) {  System.***out***.println("오답입니다.");  }  }  System.***out***.println("맞은갯수 : "+w+"/5");  }  **public** **static** **void** Subjective() {  **int** w = 0;  **int** num[] = { 0, 0, 0, 0, 0 };  **int** checknum;  **for** (**int** i = 0; i < 5; i++) {  num[i] = *r*.nextInt(*words*.size());  **for** (**int** j = 0; j < i; j++) {  **if** (num[i] == num[j]) {  i--;  **break**;  }  }  }  **for** (**int** j = 0; j < 5; j++) {  checknum = num[j];  *getWords*(checknum).upCount();  System.***out***.println(*getWords*(checknum).getWord\_e());  System.***out***.println("뜻을 입력하시요 : ");  String str = *sc*.nextLine();  str = str.trim();  **if** (*getWords*(checknum).correct(str)) {  System.***out***.println("정답입니다.");  w += 1;  } **else**  System.***out***.println("정답이 아닙니다.");  }  System.***out***.println("맞은갯수 : " + w + "/5");  }    }  //word.java  **package** konkuk;  **import** java.util.\*;  **public** **class** Word **implements** Comparable<Word>{  **static** Random *rand* =**new** Random();  **private** **int** count = 0;  **private** String word\_e;  **private** String[] mean;  **public** **void** upCount() {  **this**.count +=1;  }  **public** **boolean** zeroCount() {    **return** count==0;  }  **public** Word(String eng,String[] mean) {  word\_e=eng;  **this**.mean=mean;  }  **public** Word(String string, String string2, String string3) {  // **TODO** Auto-generated constructor stub  }  **public** String getWord\_e() {  **return** word\_e;  }  **public** String[] getMean() {  **return** mean;  }  **public** **boolean** correct(String dd) {  **for**(**int** i=0;i<mean.length;i++) {  **if**(mean[i].equals(dd))  **return** **true**;  }  **return** **false**;  }  **public** String checkMean(**int** i) {  **return** mean[i];  }  **public** String randMean() {  **return** mean[*rand*.nextInt(mean.length)];  }  @Override  **public** String toString() {    **return** "[영어 : " + word\_e + ", 뜻 :" + Arrays.*toString*(mean) + "출제수 : " +count+"]";  }  @Override  **public** **int** hashCode() {//Set으로 담을 경우- hashcode 영어로만  // **TODO** Auto-generated method stub  **return** word\_e.hashCode();  }  @Override  **public** **boolean** equals(Object obj) { //set으로 담을경우 - Word.equals는 영어만 비교  // **TODO** Auto-generated method stub  **if**(obj **instanceof** Word)  **return** word\_e.equals(((Word)obj).word\_e);  **return** **false**;  }  @Override  **public** **int** compareTo(Word o) {  // **TODO** Auto-generated method stub  **return** count - o.count;  }  } |
| **실행결과** |
|  |