Homework05\_buggy

hw05\_21300691

ECE 20016

JAVA PROGRAMMING LANGUAGE

21000498 이성진/21300691 정원식

com0673@hanmail.net/cws0173@naver.com

2014.03.28

**On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this programming assignment. Signed: 정원식**

1. Correct the compile errors. Make a note of what changes you made. You may need to make small changes and recompile to see if those errors are fixed. This may take several iterations. It is okay not to fix all the warnings but all of the non-warning problems will need to be fixed. Your program has compiled successfully. Don't try to figure out what's wrong with this line yet. We'll figure it out eventually.

* generateSpellings 메소드를 정의 할 때 반환형을 void 로 수정하였다.
* dial배열을 sigle array가 아닌 2차원 배열로 정의하였다.
* numVowel 변수를 0으로 초기화 하였다.
* phoneByte 변수의 타입을 int형으로 변경하였다.
* 변수 “word” 를 일정한 간격으로 출력시키기 위해서 println 메소드 대신 print 메소드를 사용하였고 printIfGood 함수의 return 값을 변수 “level”에 저장하였다.
* phoneString.charAt(i) - '0' 앞에다 직접 byte타입을 입력하여 (byte)(phoneString.charAt(i) - '0')구문으로 수정하였다.

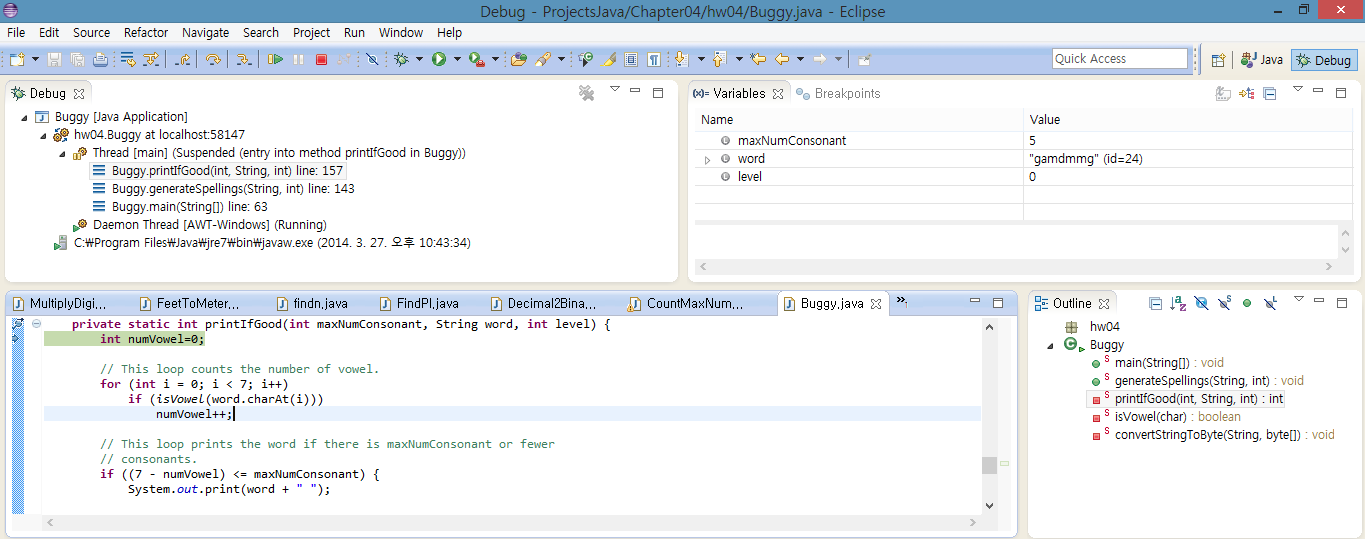
2. Using the techniques you have learned so far, Explain what phoneByte[i] = (byte) (phoneString.charAt(i) - '0') does.

phoneString.charAt(i) - '0' 에서 charAt(i)는 phoneString의 i번째 인자를 반환하는 메소드다. phoneString에는 사용자가 입력한 숫자가 char type으로 저장되어 있는데 이 데이터들은 아스키 값으로 저장되므로 i번째 char type의 아스키코드와 ‘0’ 아스키코드간에 –(마이너스) 연산을 수행할 수 있다. 연산의 결과는 default인 int형으로 반환이 되는데 byte로 선언된 phoneByte변수에 저장하기 위해서 phoneString.charAt(i) - '0' 식 앞에 (byte)를 붙여 byte로 type을 변환시켜주어야 한다.

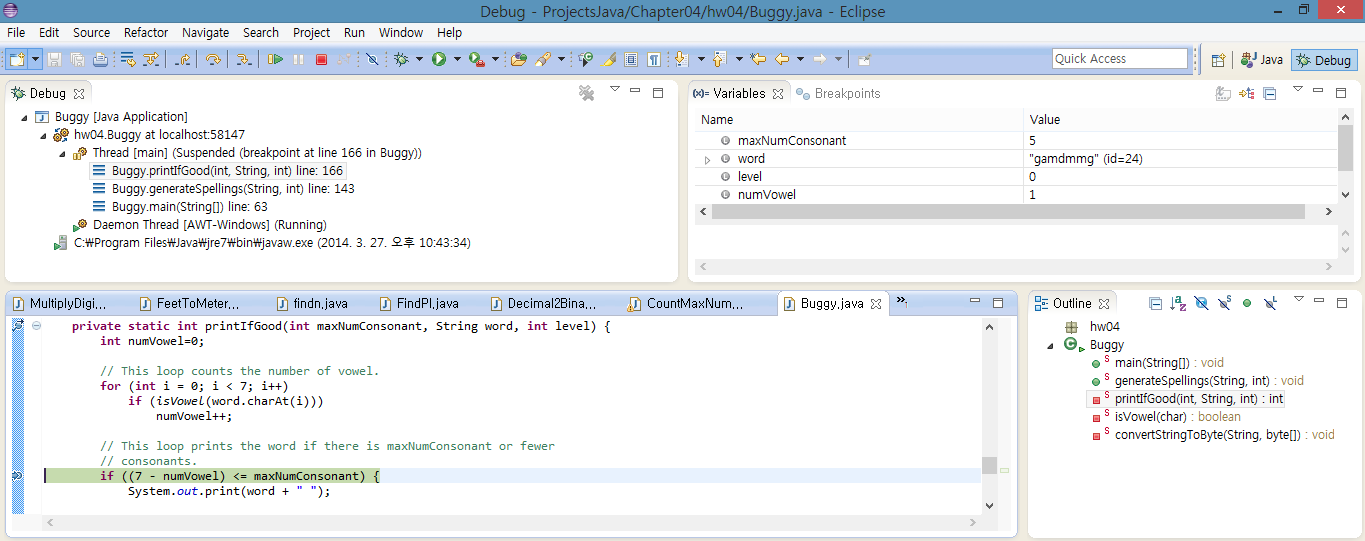
3. After calling convertStringToByte method, explain what values are in phoneByte variable.

convertStringToByte 메소드는 phoneString안에 char형으로 저장되어 있는 숫자들을 char ‘0’의 아스키코드와의 차를 구하여 phoneByte변수 안에 저장하는 기능을 수행한다. 예를 들면 phoneString에 4가 저장되어 있다고 할 때 저장되어있는 ‘4’ – ‘0’ 연산을 수행하면 아스키코드인 84 - 80이 계산되어 숫자 4를 얻을 수 있다. 이 값은 byte형으로 변환되어 phoneByte변수에 저장된다.

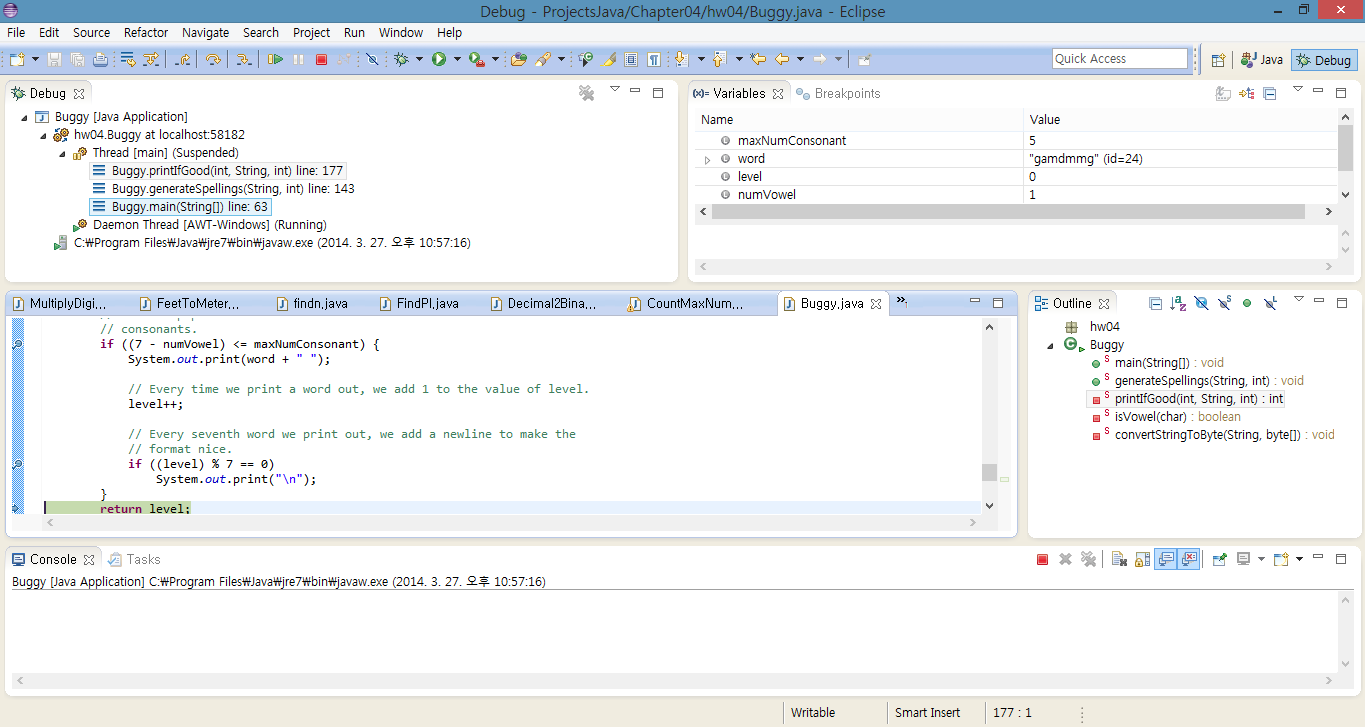
4. Capture a screen shot of Debug window, after calling printIfGood method and explain it. What are the argument values when printIfGood() is invoked at the very first time with user’s inputs of (5 consonants and 4263664 handong)?



처음 메소드를 불렀을 때, argument value들은 maxNumConsonant: 5, word: gamdmmg, level: 0 이다.



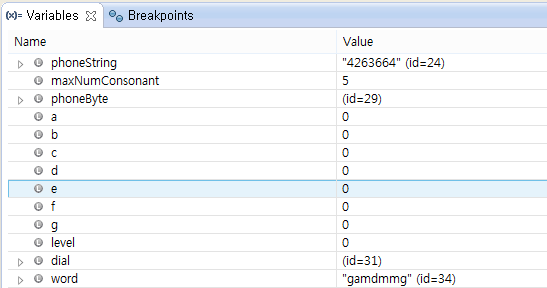
이후 첫번째 줄의 int형 변수인 numVowel을 초기화하고 다음에 나오는 for문 실행시 word의 각 단어의 모음 여부를 확인한다. 그 결과 numVowel은 1이 증가되어 저장된다.



이후 if문을 실행한다. if문에서 vowel의 값으로 자음의 수를 구하여 maxNumConsonant와 비교하여 작거나 같을시 출력하고 level의 값을 1올린다. 또 마지막 if문에서 level이 7로 나눠질시, 개행을 한다. 그 결과 gamdmmg는 조건을 충족시키지 못하여 출력되지 않았다.

5,6. Starting on the above line, a series of nested loops is encountered. How does each nested loop work? Make sure you step through the loop and demonstrate that you've actually examined how the loop works.

우선 위 루프를 한 번 실행하였다.



그 결과 변수 값이 위 그림과 같이 나왔다. 그 후 level=*printIfGood*(maxNumConsonant, word, level); 문장을 실행한 결과 word는 조건을 만족하지 못하여 출력되지 않았고 level의 값은 증가하지 않았다. 가장 가까운 루프가 반복되므로 g = 1 로 g 값이 1 증가하였다. a,b,c,d,e,f = 0이고 g = 1 일때, word는 gamdmmh이 저장된다. g = 2이고 g < 3 조건에 만족하므로 루프는 한번 더 실행되고 a,b,c,d,e,f = 0이고 g = 2 이므로, word에는 gamdmmi이 저장된다. g = 3 으로 증가하면 더 이상 루프가 돌아가지 않고 그 다음으로 가까운, 다시 말해서 바로 상위에 있는 루프가 수행된다. f = 1 로 증가되었고 다시 g = 0으로 초기화되어 루프가 실행되어 이 때 a,b,c,d,e = 0, f = 1, g = 0 이므로 word에는 gamdmng이 저장된다. 각 반복문의 maxNumConsonant, word, level은 printIfGood 함수 안으로 복사되어 함수가 수행된다.

즉, 하위 루프가 돌아가면서 하위 루프의 조건식을 만족하지 못 할 때마다 상위 루프를 돌리고 결국 최상위 루프가 조건식을 만족하지 못하게 되면 반복문은 종료가 된다.

nested loop들은 word를 모든 경우로 조합시키며 printIfGood 메소드에서 word가 조건을 충족하는 경우 출력되고 level의 값을 증가시킨다. 각각의 nested loop들은 3번 돌리며 loop의 수가 7개이므로 3^7개의 조합을 가능하게 한다.

7. What is role of printIfGood method? What is the very first ***word*** that printIfGood() prints with user’s inputs of (5 consonants and 4263664 handong)?

스트링 변수 word에 저장되어 있는 각 문자들의 모음여부를 판단한다. 모음이 한 문자 판별될 때마다 numVowel 숫자값을 1씩 증가시키고 총 문자수인 7에서 모음 갯수를 빼서 얻는 자음 갯수를 사용자가 입력한 최대 자음 갯수와 비교한다. 사용자가 입력한 최대 자음 갯수를 만족하는 word의 문자열만 출력한다. 그리고 level 을 체크하여, 즉 word가 7번 출력될 때마다 엔터키를 삽입한다. 유저가 5개의 모음과 4263664 를 입력했을 때 printIfGood()에서 출력되는 첫 번째 word 는 “gamdmmi” 이다.

8. What is the role of level variable?

word가 사용자가 입력한 최대 자음 갯수를 만족하여 값이 출력할 때마다 level 변수가 count 된다. (level) % 7 == 0 식을 통하여 word가 출력되는 7번째마다 if 구문이 실행되어 엔터값이 들어 간다. 즉 한 줄에 word값이 7개씩 출력되도록 한다. level에는 word변수가 출력될 때마다 count된 개수가 저장된다.

9. Why the printIfGood method is returning the value of level variable?

level는 최대 모음개수를 만족한 word가 출력될 때마다 1씩 증가된다. 그런데 printIfGood함수 안에서 증가된 level값이 return되지 않고 함수가 끝나버린다면 증가된 level값은 printIfGood함수 안에서만 존재했을 뿐 함수 밖으로 나오는 순간 사라지게 되어 증가한 값을 누적하여 저장할 수 없다. 그래서 level 값을 int형으로 return하여 return값을 generateSpellings 함수의 level에 저장한다. 이렇게 한다면 printIfGood함수 안에서 증가한 level값을 그대로 유지하고 값을 누적해나갈 수 있다.

10. Finally, explain what the problem is in this section of code. Fix the problem in the code, and recompile the program .

* generateSpellings함수의 return type이 정의되지 않아서 void type으로 반환되도록 public static void generateSpellings 으로 수정하였다.
* dial 배열은 2차원 배열로 사용되는데 1차원 배열로 선언되어서 char[][] dial 중괄호 하나를 더 입력하여 2차원 배열이 되도록 수정하였다.
* word에 모음이 저장되어 있을 때 numVowel을 1부터 1씩 증가시키기 위해서 numVowel을 0으로 초기화 하였다.
* “word”에 저장되어 있는 문자를 한 줄에 7단어씩만 출력되게 하기 위해서 System.out.println(word + " ");의 println 메소드를 print 메소드로 바꾸어 주었다. 그리고 printIfGood에서 return되는 level값이 저장되어 누적되도록 level=printIfGood(maxNumConsonant, word,level);와 같이 printIfGood함수의 반환값을 level 변수에 저장하는 구문을 수정하였다.
* phoneByte[i] = (phoneString.charAt(i) - '0'); 이 구문에서 phoneString.charAt(i) - '0' 의 계산은 default 타입인 int type으로 계산이 된다. phoneByte는 byte type의 배열이므로 int type보다 크기가 작은 byte type에 입력시키기 위하여 phoneByte[i] = (byte)(phoneString.charAt(i) - '0'); 구문에 (byte)를 입력하였다.
* printIfGood함수에서 int type인 level변수의 값이 return 되는데 *printIfGood*(maxNumConsonant, word, level); 이 구문만으로는 c와 달리 포인터를 통해 return 값을 저장하는 기능을 수행하지 못한다. 그래서 printIfGood 함수에서 return 되는 level값을 저장할 수 있도록 level=*printIfGood*(maxNumConsonant, word,level); 이와 같이 구문을 구성한다.