

로. 여기서는 사용할 수가 없다.

put() 메소드는 get() 메소드와 동일한 초기 조치를 수행한다. 먼저 주어진 key에 대해 배열을 탐색한다. 값이 발견되면, 중첩된 Entry 클래스의 setValue() 메소드를 호출하여 항목의 현재 value를 명시된 value로 대체하고, 옛날 value 값을 리턴한다. 값이 발견되지 않으면, 명시된 (key, value) 쌍에 대해 새로운 Entry 객체를 생성하고 이것을 배열의 다음 가용 위치에 삽입한다.

라인 16에서 size 카운터의 값이 배열에 대한 인덱스로 사용된 후에 1만큼 증가됨에 유의하라. 이러한 방식으로 size는 배열에서 다음 가용 위치에 대한 인덱스와 같아지게 되며, 이 값은 항상 해당 위치 앞에 있는 원소의 수와 같게 된다. 예를 들어, size가 4이면 a[4]가 다음 가용 위치가 되며, 이는 이미 사용 중인 원소들이 a[0], a[1], a[2], a[3]이기 때문이다.

toString() 메소드는 라인 34에서 Entry.toString() 메소드를 사용한다. 이 메소드는 문자열 수식 a[i]+“\n”에서 자동으로 호출되는데, 그 이유는 a[i]가 Entry 객체이고 “\n”이 String 상수이기 때문이다.

Main 클래스는 다음의 연산들을 수행한다.

1. 텍스트 문서를 한 줄씩 읽어가면서 각 단어를 파싱한다.
2. 단어가 맵에 없으면, 그 단어와 그 라인 번호를 맵에 집어 넣는다.
3. 단어가 이미 맵에 있으면, 그 라인 번호를 해당 단어의 리스트에 추가한다.
4. 전체 맵을 프린트한다.

이 알고리즘은 리스팅 3.18에 구현되어 있다.

리스팅 3.18 ArrayMap 클래스의 테스트

```

1  import java.io.*; // defines the FileReader and BufferedReader classes
2  import java.util.StringTokenizer;
3
4  public class Main {
5
6      public Main(String file) {
7          Map map = new ArrayMap();
8          int lineNumber = 0;
9          try {
10             BufferedReader in = new BufferedReader(new FileReader(file));
11             String line = in.readLine();
12             while(line != null) {
13                 ++lineNumber;
14                 StringTokenizer parser = new StringTokenizer(line, " ,;:-.?!");
15                 while (parser.hasMoreTokens()) {

```