가장 흔한 단어

문제 : 금지된 단어를 제외하고 가장 흔하게 등장하는 단어를 출력하라. 대소문자를 구분하지 않으며, 마침표, 쉼표 또한 무시한다.

예시1)
Input: paragraph = "Bob hit a ball, the hit BALL flew far after it was hit.", banned = ["hit"]
Output: "ball"

입력값에는 대소문자가 섞여 있으며, 쉼표, 마침표가 존재한다. 따라서 데이터 클랜징이라 부르는 입력값에 대한 전처리 작업이 필요하다. 좀 더 편리하게 처리하기 위해서는 정규식을 사용하면 된다.

```
String[] words = p.replaceAll("\\W+", " ").toLowerCase().split(" ");
```

정규식에서 `\W`는 단어 문자가 아닌 것을 뜻한다.
참고로 단어 문자를 뜻 할때는 `\w` 소문자 w이다. 뒤에 +를 붙이면 연속적인 값을 의미한다.
따라서 위 `p.replaceAll("\\W+", " ")`는 단어 문자가 아닌 값을 전부 " "공백으로 치환한다는 의미이다.
제약 조건으로 대소문자를 구분하지 않는다고 했음으로 toLowerCase()로 모두 소문자로 바꿔주며, split(" ")을 써서 띄어쓰기 기준으로 단어를 분리한다.

이젠 금지된 단어를 제외하고 각 단어가 몇 차례나 등장하는지 개수를 세어보자

```
for (String w : words){
        if(!ban.contains(w)) {
                counts.put(w, counts.getOrDefault(w, 0) + 1);
        }
}
```

여기서 getOrDefault() 메서드는 값이 존재하지 않는 경우 기본값을 출력하며, 존재하는 경우 해당하는 값을 출력한다. 여기에 +1을 하고 다시 저장한다.

```
Map<String, Integer> counts = new HashMap<>();
 ...
Collections.max(counts.entrySet(),
Map.Entry.comparingByValue()).getKey()
```

Collections.max()는 가장 큰 값을 찾는다. counts.entrySet()에서 저장된 가장 큰 값을 찾으면 아래와 같이 나온다.

```
[the=1, a=1, ball=2, away=1, far=1, flew=1, ross=1, was=1, after=1,
it=1]
```

여기서 가장 큰 값을 찾으며, 찾는 기준은 Map.entry.comparingByValue()이다. HashMap으로 저장된 값중 value가 가장 큰 것을 찾고 getKey()를 통해 키 값만 가져오게된다.

```
public String mostCommonWord(String p, String[] banned){
        // 금지어 목록이 String 배열이므로, 비교 메서드를 제공하는 Set으로 변경한다.
        Set<String> ban = new HashSet<>(Arrays.asList(banned));
        // 각 단어별 개수가 저장될 Map
        Map<String, Integer> counts = new HashMap<>();

        // 전처리 작업 후 단어 목록을 배열로 저장
        String[] words = p.replaceAll("\\W+", " ").toLowerCase().split("
");

        for (String w : words){
                // 금지된 단어가 아닌 경우 개수 처리
                if (!ban.contains(w)){
                        // 존재하지 않는 단어라면 기본값을 0으로 지정, 추출한 값에
+ 1 하여 저장
                        counts.put(w, counts.getOrDefault(w, 0) + 1);
                }
        }
        // 가장 흔하게 등장하는 단어 추출
        return Collections.max(counts.entrySet(),
Map.Entry.comparingByValue()).getKey();
}
```