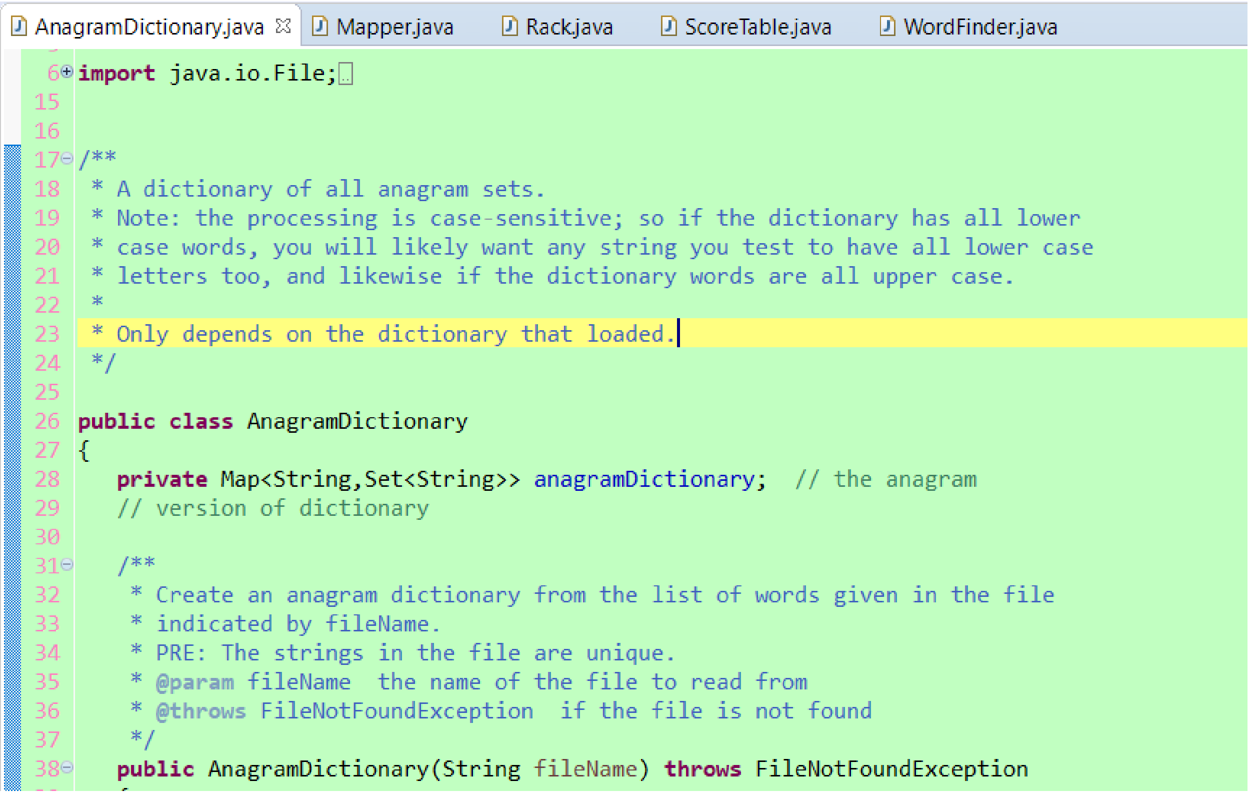
Reminder: this document is supposed to be confidential. Please do not show this to anyone else.

Word Bank for a game

Store a dictionary for the game. Search candidate word from the dictionary to win the word game.



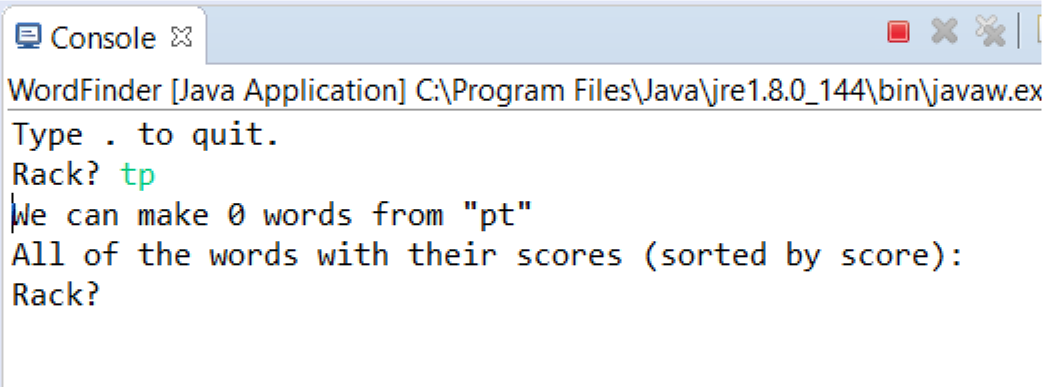
```java
import java.io.File;


/**
 * A dictionary of all anagram sets.
 * Note: the processing is case-sensitive; so if the dictionary has all lower
 * case words, you will likely want any string you test to have all lower case
 * letters too, and likewise if the dictionary words are all upper case.
 *
 * Only depends on the dictionary that loaded.
 */

public class AnagramDictionary
{
   private Map<String,Set<String>> anagramDictionary;  // the anagram
   // version of dictionary

   /**
     * Create an anagram dictionary from the list of words given in the file
     * indicated by fileName.
     * PRE: The strings in the file are unique.
     * @param fileName  the name of the file to read from
     * @throws FileNotFoundException  if the file is not found
     */
   public AnagramDictionary(String fileName) throws FileNotFoundException
```

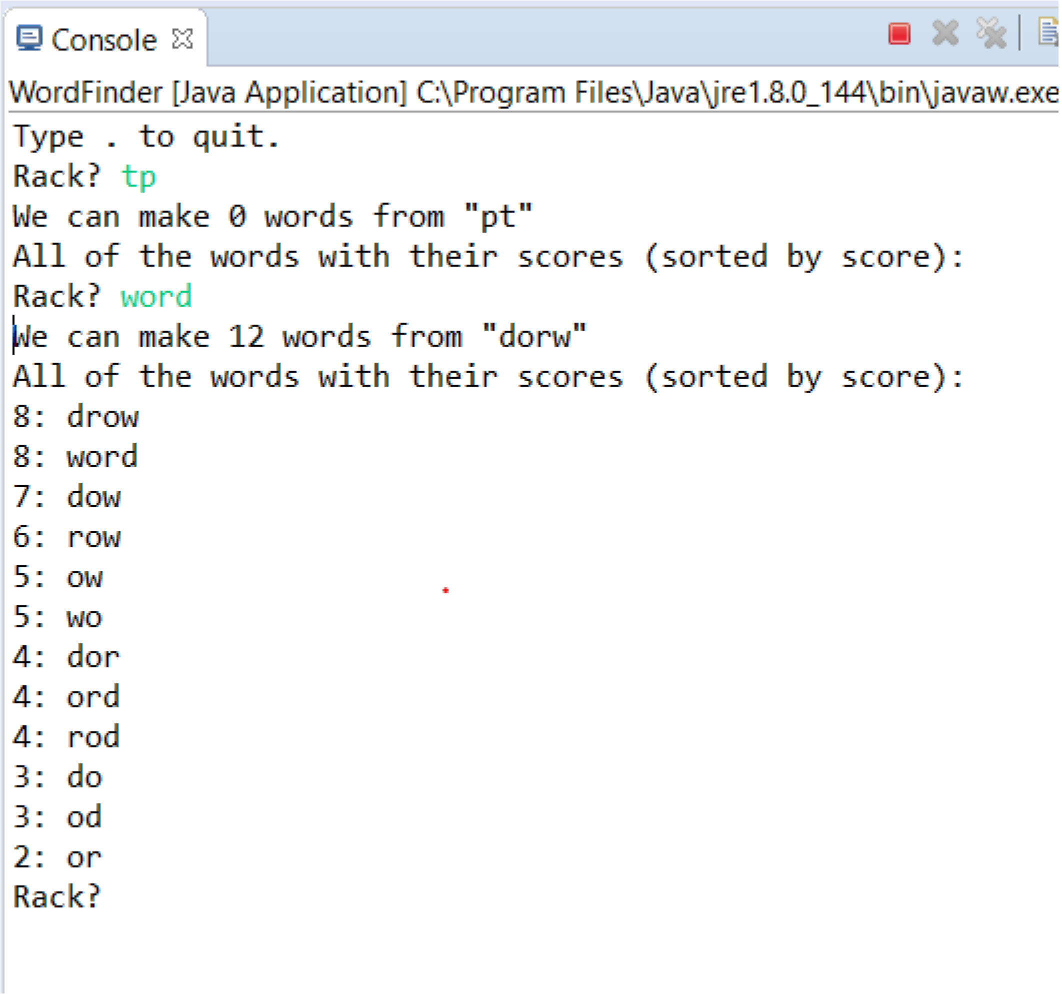
1. Search word using letters “tp”:



```
WordFinder [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.ex
Type . to quit.
Rack? tp
We can make 0 words from "pt"
All of the words with their scores (sorted by score):
Rack?
```

Reminder: this document is supposed to be confidential. Please do not show this to anyone else.

2. Search word with letters “word”:



WordFinder [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe

```
Type . to quit.
Rack? tp
We can make 0 words from "pt"
All of the words with their scores (sorted by score):
Rack? word
We can make 12 words from "dorw"
All of the words with their scores (sorted by score):
8: drow
8: word
7: dow
6: row
5: ow
5: wo
4: dor
4: ord
4: rod
3: do
3: od
2: or
Rack?
```