Purpose	Practice using a Map
What to Submit	Submit your code to a project named Lab6 on Bitbucket.

1. Implement a WordCounter class

Create a class that count how many times different words occur. Each time the addWord() method is called, you add to the occurrences for that word. Ignore the case of words in counting words.

The class should have these methods:

void addWord(String word)	Add one to the count of occurrences for this word, ignoring case.
Set<String> getWords( )	Get all the words seen so far, as a Set.
int getCount(String word)	Get the number of occurrences for a given word.
String[ ] getSortedWords( )	Get all the words seen so far, sorted in alphabetical order.

2. Create a WordCount class to Read a File

Create a WordCount class with a main method to read words from a file and use the WordCounter to count the words.

2.1 Print the first 20 words (in alphabetical order) and the number of times each word occurs in the text.

2.2 Print the 20 *most frequently used word*s in the text. You need to (somehow) sort the words.

Use this file (except from *Alice in Wonderland*) from the class week6 folter:

https://bitbucket.org/skeoop/oop/raw/master/week6/Alice-in-Wonderland.txt

You can download the file to your PC and open it as a FileInputStream (or FileReader).

Or, read it directly from the Internet as a URL:

```
String FILE_URL =
        "https://bitbucket.org/skeoop/oop/raw/master/week6/Alice-in-Wonderland.txt";
URL url = new URL( FILEURL);
InputStream input = url.openStream( );
Scanner scanner = new Scanner( input );
```

How to Remove Punctuation?

The text for *Alice in Wonderland* contains punctuation, such as:

She said, "Oh dear, what shall I do now?"

We want to split the text into words every time a *whitespace* (space, tab, newline) or *punctuation* mark occurs. The Scanner class lets you specify a set of delimiter characters as a *regular expression*. A regular expression for the delimiters in Alice in Wonderland is:

```
final String DELIMS = "[\\s,.\\?!\"():;]+";
scanner.useDelimiter( DELIMS );
```

Regular expressions are standard and available in most programming languages. The Java Tutorial has a section on regular expressions.