Programming Assignment 9-2

If you use the “SymbolBalancer” class, then most of the start-up code for this exercise (described below) has already been written for you. If you use this class, then you will only need to implement the checkDelimiters() method. Note: Using the “SymbolBalancerMin” class will be more of a challenge.

The startup code has a class `SymbolBalancer` that has a constructor

```
   SymbolBalancer(String filename)
```

which accepts the name of a file to examine, and that also has a method

```
      boolean symbolsBalanced(String delimiters)
```

The delimiters argument is a list of all pairs of delimiters that will be used by your symbol balanced method. For example, here is a possible value of the delimiters parameter:

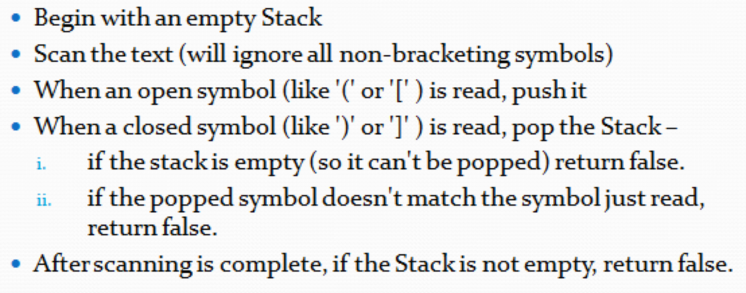
```
   "[](){}"
```

The `String` that is passed into this argument must be parsed. You can do this in a loop with repeated calls to `charAt`.

Also, there is an instance variable `text` that stores the text to be parsed (which you will extract from the input file).

Your method `symbolsBalanced` should return true if the open/closed pairs of delimiters specified in the delimiters argument, as they occur in the text that is being examined, are balanced; false, otherwise. To accomplish this, use the following procedure, as described in the slides:

Procedure for Checking Delimiter Balance



- Begin with an empty Stack
- Scan the text (will ignore all non-bracketing symbols)
- When an open symbol (like '(' or '[') is read, push it
- When a closed symbol (like ')' or ']') is read, pop the Stack –
  - i. if the stack is empty (so it can't be popped) return false.
  - ii. if the popped symbol doesn't match the symbol just read, return false.
- After scanning is complete, if the Stack is not empty, return false.

Additional Information:

We will explain more formally how to read a file in a later lesson. For now, the code just uses the readFile method, shown here:

```
void readFile() {
   final String ODD_DELIMITER = ""+(char)0;
   try {
      Scanner sc = new Scanner(new File(filename));
      sc.useDelimiter(ODD_DELIMITER);

      //gets the next "token", which happens to be
      //all of the text in the file (because of the odd delimiter)
      text = sc.next();
      System.out.println(text);
      sc.close();
   }
   catch(FileNotFoundException ex) {
      System.err.println("File Not Found Exception
                         "+ex.getMessage());
   }
}
```

You can test your symbol-balanced-checking code in a `main` method by reading in the `Employee.java` class (provided in a folder in this directory) using the `readFile()` method. (See the `readme` file for instructions about where to place the input file in your directory.) In your call to `symbolsBalanced()`, pass in the following `String` of delimiter pairs: "`[]{}<>()||`". Your `main` method should simply output either "true" or "false" to the console, indicating the result of the symbol-balanced test.