

The ex45 Case Study (Part 2): the Driver Class & Runtime Behavior

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

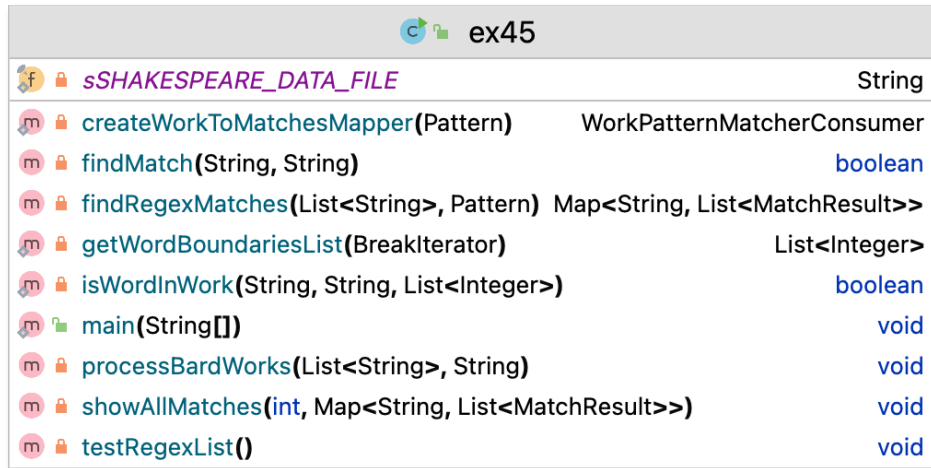
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of the ex45 case study that searches for words & phrases in the works of Shakespeare
- Recognize the design, implementation, & runtime behavior of the ex45 driver



Method Signature	Return Type
<code>sSHAKESPEARE_DATA_FILE</code>	String
<code>createWorkToMatchesMapper(Pattern)</code>	WorkPatternMatcherConsumer
<code>findMatch(String, String)</code>	boolean
<code>findRegexMatches(List<String>, Pattern)</code>	Map<String, List<MatchResult>>
<code>getWordBoundariesList(BreakIterator)</code>	List<Integer>
<code>isWordInWork(String, String, List<Integer>)</code>	boolean
<code>main(String[])</code>	void
<code>processBardWorks(List<String>, String)</code>	void
<code>showAllMatches(int, Map<String, List<MatchResult>>)</code>	void
<code>testRegexList()</code>	void

```
ex45 [ex45.main()] ×
Number of works that match is 13 out of a total of 37
The Tragedy of Othello, Moor of Venice
"lord, your true" [131546]
"lord? with whom? how am I false" [131908]
The Tragedy of Titus Andronicus
"lord, sith true" [15533]
The History of Troilus and Cressida
"lord, will you be true" [120358]
The Taming of the Shrew
"LORD. 'Tis very true" [5568]
The Tragedy of King Lear
"lord, and true" [6392]
The Tragedy of Julius Caesar
"lord, are false" [108997]
Cymbeline
"LORD. [Aside] If it be a sin to make a true" [12336]
The Tragedy of Hamlet, Prince of Denmark
"lord, 'tis true" [20976]
The First Part of King Henry IV
```

See github.com/douglasraigschmidt/LiveLessons/tree/master/Java8/ex45

Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of the ex45 case study that searches for words & phrases in the works of Shakespeare
- Recognize the design, implementation, & runtime behavior of the ex45 driver
 - Again, this class uses the Java streams framework extensively



See www.oracle.com/technical-resources/articles/java/architect-streams-pt2.html

Runtime Behavior of the ex45 Class

Runtime Behavior of the ex45 Class

- Test a utility method that converts a List of words into a regular expression

```
private void testRegexList() {  
    var quote = ""  
        The quick brown fox jumps  
        \nover the lazy dog.""";  
  
    var wordsToMatch = List  
        .of("Cat", "Dog",  
            "Mouse", "sox");  
  
    var regexString =  
        makeRegex(wordsToMatch);  
  
    ...  
}
```

Runtime Behavior of the ex45 Class

- Test a utility method that converts a List of words into a regular expression
- Read the complete works of William Shakespeare from a text file & store them in a List

```
List<String> bardWorks =  
    BardDataFactory.getInput  
        (sSHAKESPEARE_DATA_FILE,  
         "@");  
The Sonnets  
...  
@The Tragedy of Hamlet  
...  
@The Tragedy of Julius Caesar  
...  
@The Tragedy of Macbeth  
...
```

Runtime Behavior of the ex45 Class

- Test a utility method that converts a List of words into a regular expression
- Read the complete works of William Shakespeare from a text file & stores them in a List
- Search for occurrences of a specific word ("lord" in this case) in the complete works of Shakespeare
 - Filter the list to only include works containing word & compile a regex pattern for further matching

```
var bardWorksMatchingWord =  
    bardWorks  
        .stream()  
        .filter(work ->  
            findMatch(work, word));  
    .toList();
```

```
var pattern = Stream  
    .of("\\b" + word + "\\b" +  
        makeRegex(List.of("true",  
                            "false"))  
    .map(regex -> Pattern  
        .compile(regex,  
            CASE_INSENSITIVE))  
    .findFirst().orElse(null);
```

Runtime Behavior of the ex45 Class

- Test a utility method that converts a List of words into a regular expression
- Read the complete works of William Shakespeare from a text file & stores them in a List
- Search for occurrences of a specific word ("lord" in this case) in the complete works of Shakespeare
- Take the filtered works & a compiled regex pattern to find & display all the matching lines for each work

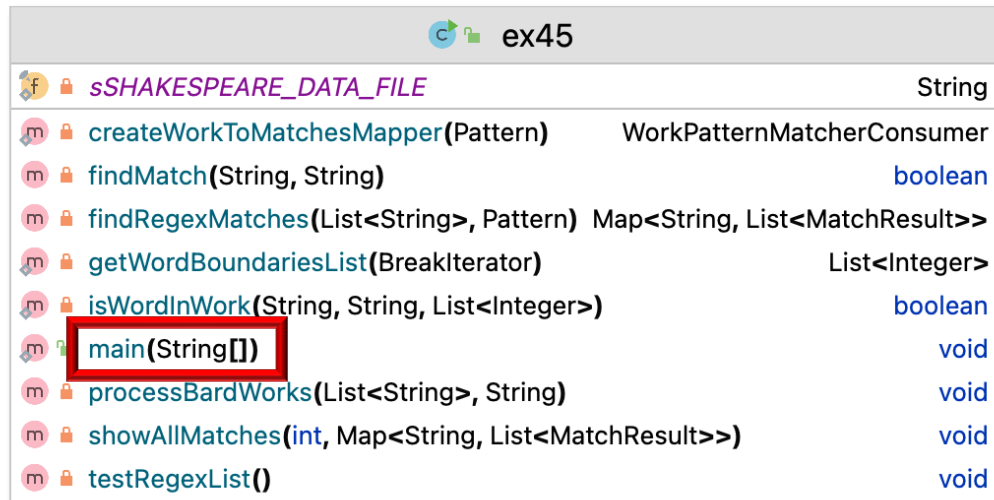
```
ex45 [ex45.main()] ×
Number of works that match is 13 out of a total of 37
The Tragedy of Othello, Moor of Venice
"lord, your true" [131546]
"lord? with whom? how am I false" [131908]
The Tragedy of Titus Andronicus
"lord, sith true" [15533]
The History of Troilus and Cressida
"lord, will you be true" [120358]
The Taming of the Shrew
"LORD. 'Tis very true" [5568]
The Tragedy of King Lear
"lord, and true" [6392]
The Tragedy of Julius Caesar
"lord, are false" [108997]
Cymbeline
"LORD. [Aside] If it be a sin to make a true" [12336]
The Tragedy of Hamlet, Prince of Denmark
"lord, 'tis true" [20976]
The First Part of King Henry IV
```

Methods in the ex45 Class

Methods in the ex45 Class

```
void main(String[] argv)
```

- Serves as the program's main entry point
- Reads Shakespeare's complete works into a List of Strings
- Calls the methods `testRegexList()` & `processBardWorks()` to perform various text analyses



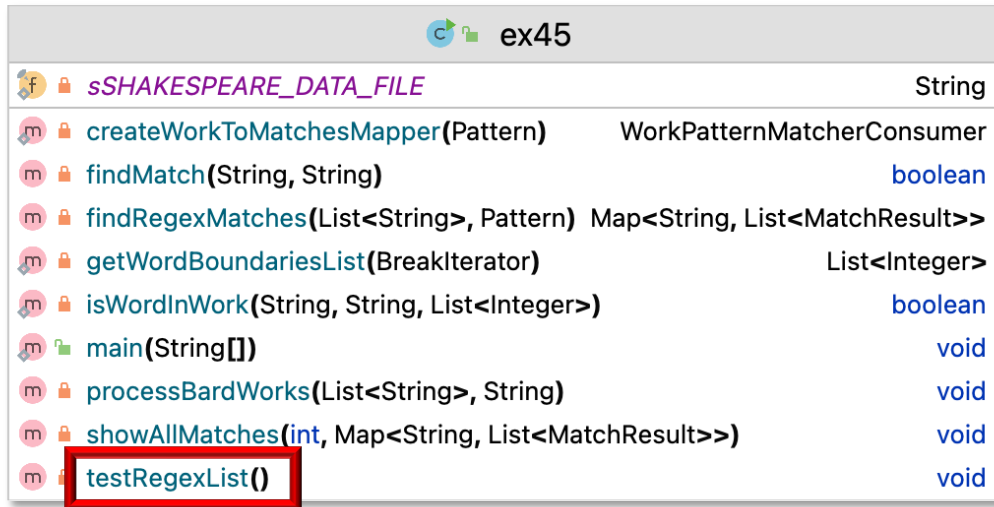
Icon	Method Name	Return Type
f	<code>sSHAKESPEARE_DATA_FILE</code>	String
m	<code>createWorkToMatchesMapper(Pattern)</code>	WorkPatternMatcherConsumer
m	<code>findMatch(String, String)</code>	boolean
m	<code>findRegexMatches(List<String>, Pattern)</code>	Map<String, List<MatchResult>>
m	<code>getWordBoundariesList(BreakIterator)</code>	List<Integer>
m	<code>isWordInWork(String, String, List<Integer>)</code>	boolean
m	<code>main(String[])</code>	void
m	<code>processBardWorks(List<String>, String)</code>	void
m	<code>showAllMatches(int, Map<String, List<MatchResult>>)</code>	void
m	<code>testRegexList()</code>	void

See [Java8/ex45/src/main/java/ex45.java](https://github.com/jvanderhaeghe/java8-exercises/blob/master/ex45/src/main/java/ex45.java)

Methods in the ex45 Class

```
void testRegexList()
```

- Tests the RegexUtils makeRegex() utility method
- Constructs a regular expression from a list of words
- Checks if a quote matches the constructed regular expression & prints the result



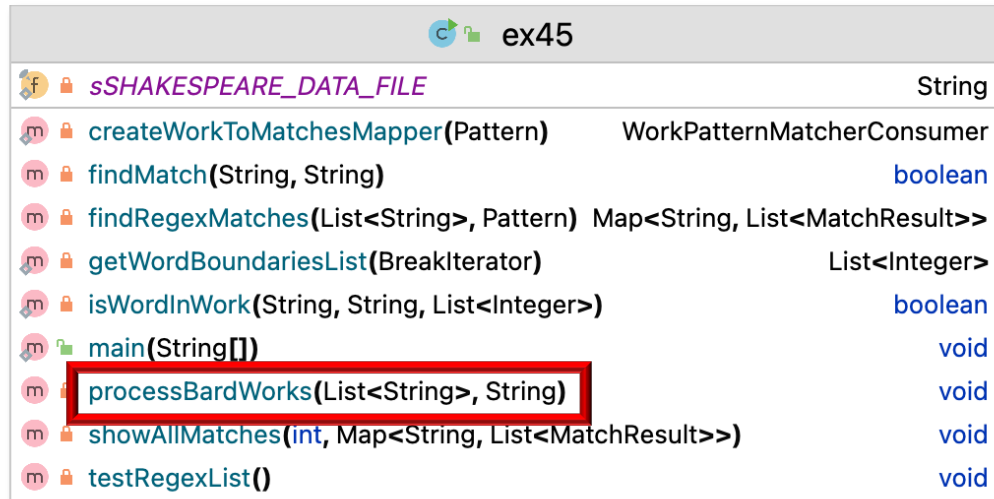
The screenshot shows the class hierarchy for the 'ex45' class. The methods listed are:

Method Name	Return Type
<code>sSHAKESPEARE_DATA_FILE</code>	String
<code>createWorkToMatchesMapper(Pattern)</code>	WorkPatternMatcherConsumer
<code>findMatch(String, String)</code>	boolean
<code>findRegexMatches(List<String>, Pattern)</code>	Map<String, List<MatchResult>>
<code>getWordBoundariesList(BreakIterator)</code>	List<Integer>
<code>isWordInWork(String, String, List<Integer>)</code>	boolean
<code>main(String[])</code>	void
<code>processBardWorks(List<String>, String)</code>	void
<code>showAllMatches(int, Map<String, List<MatchResult>>)</code>	void
<code>testRegexList()</code>	void

Methods in the ex45 Class

`void processBardWorks ()`

- Searches Shakespeare's complete works for a specific word
- Utilizes Java Streams & regular expressions for text processing
- Invokes `findMatch()` to filter works, `findRegexMatches()` to find all the matches, & `showAllMatches()` to display matching portions

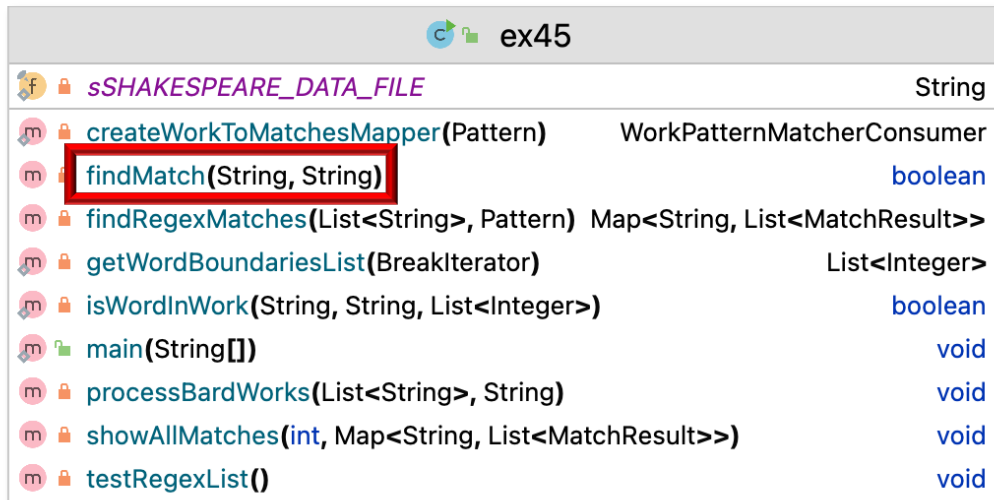


```
ex45
String
sSHAKESPEARE_DATA_FILE
createWorkToMatchesMapper(Pattern) WorkPatternMatcherConsumer
findMatch(String, String) boolean
findRegexMatches(List<String>, Pattern) Map<String, List<MatchResult>>
getWordBoundariesList(BreakIterator) List<Integer>
isWordInWork(String, String, List<Integer>) boolean
main(String[]) void
processBardWorks(List<String>, String) void
showAllMatches(int, Map<String, List<MatchResult>>) void
testRegexList() void
```

Methods in the ex45 Class

`boolean findMatch()`

- Accepts a work of Shakespeare & a search word
- Utilizes BreakIterator to identify word boundaries in the work
- Returns true if the search word exists in the work, otherwise returns false

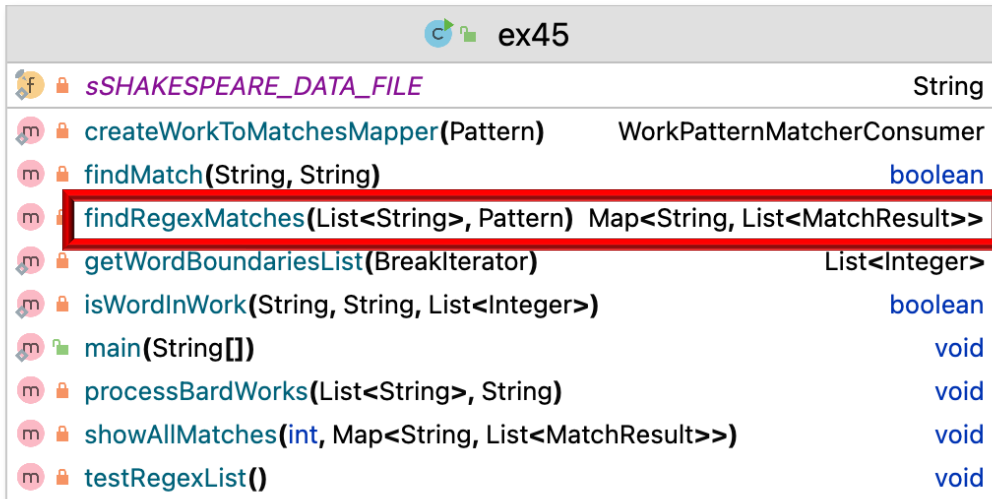


```
ex45
String
sSHAKESPEARE_DATA_FILE
createWorkToMatchesMapper(Pattern) WorkPatternMatcherConsumer
findMatch(String, String) boolean
findRegexMatches(List<String>, Pattern) Map<String, List<MatchResult>>
getWordBoundariesList(BreakIterator) List<Integer>
isWordInWork(String, String, List<Integer>) boolean
main(String[]) void
processBardWorks(List<String>, String) void
showAllMatches(int, Map<String, List<MatchResult>>) void
testRegexList() void
```

Methods in the ex45 Class

`findRegexMatches ()`

- Receives a List containing all the works of Shakespeare & a compiled regular expression pattern
- Identifies portions of text in the works that match the pattern
- Collects & returns these as a Map

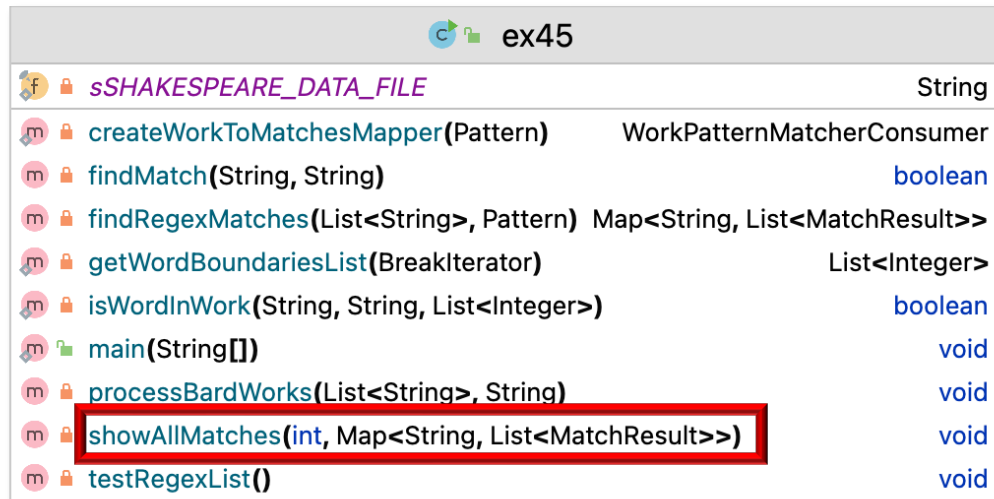


```
ex45  
f sSHAKESPEARE_DATA_FILE String  
m createWorkToMatchesMapper(Pattern) WorkPatternMatcherConsumer  
m findMatch(String, String) boolean  
m findRegexMatches(List<String>, Pattern) Map<String, List<MatchResult>>  
m getWordBoundariesList(BreakIterator) List<Integer>  
m isWordInWork(String, String, List<Integer>) boolean  
m main(String[]) void  
m processBardWorks(List<String>, String) void  
m showAllMatches(int, Map<String, List<MatchResult>>) void  
m testRegexList() void
```

Methods in the ex45 Class

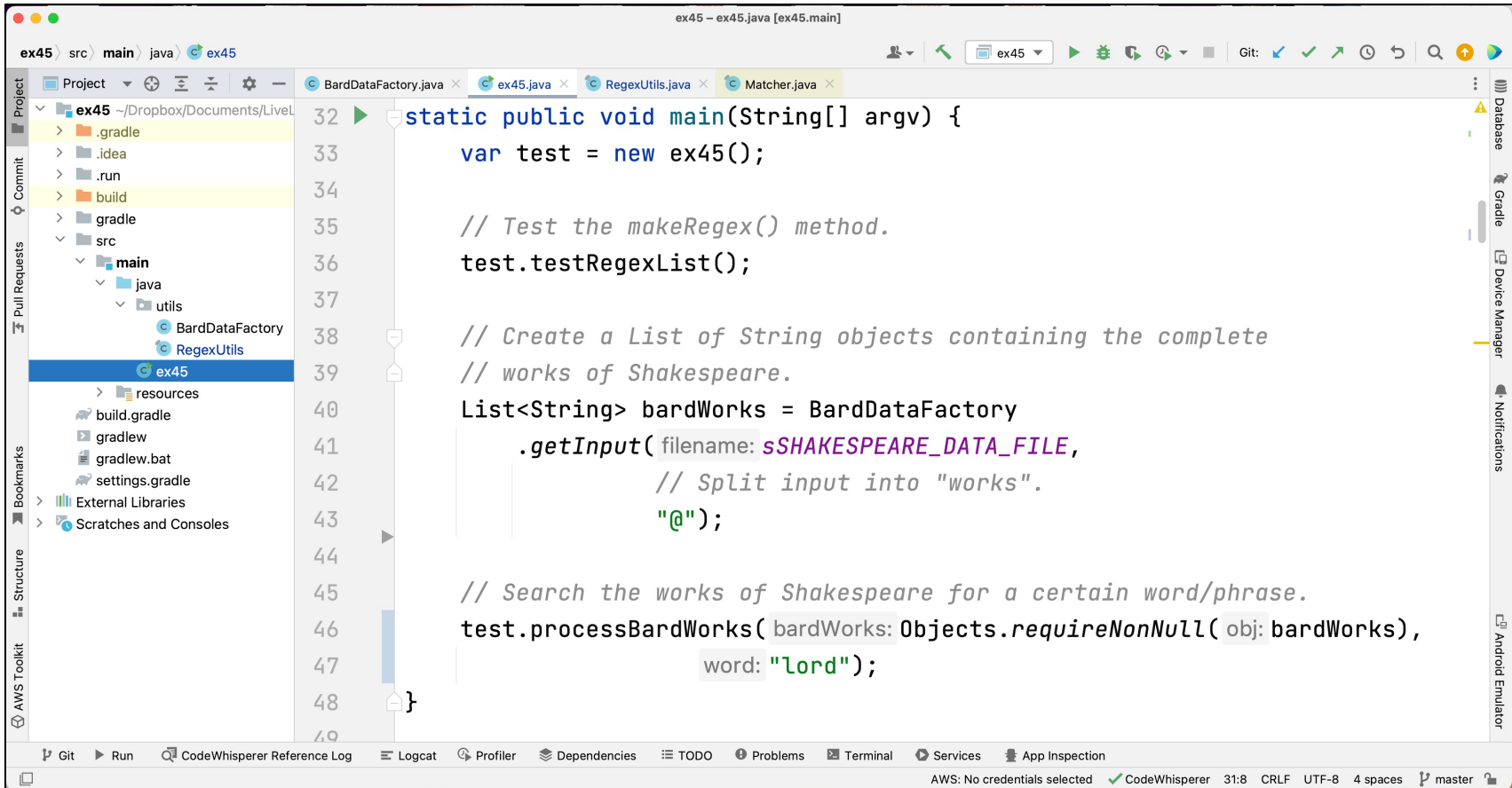
showAllMatches ()

- Prints the Map of matches that are organized by the title of each work



Icon	Method Name	Return Type
f	sSHAKESPEARE_DATA_FILE	String
m	createWorkToMatchesMapper(Pattern)	WorkPatternMatcherConsumer
m	findMatch(String, String)	boolean
m	findRegexMatches(List<String>, Pattern)	Map<String, List<MatchResult>>
m	getWordBoundariesList(BreakIterator)	List<Integer>
m	isWordInWork(String, String, List<Integer>)	boolean
m	main(String[])	void
m	processBardWorks(List<String>, String)	void
m	showAllMatches(int, Map<String, List<MatchResult>>)	void
m	testRegexList()	void

Methods in the ex45 Class



```
ex45 - ex45.java [ex45.main]
ex45 > src > main > java > ex45
Project: ex45 - ~/Dropbox/Documents/Level...
  > .gradle
  > .idea
  > .run
  > build
  > gradle
  > src
    > main
      > java
        > utils
          BardDataFactory
          RegexUtils
          ex45
      > resources
    build.gradle
    gradle
    gradlew.bat
    settings.gradle
  > External Libraries
  > Scratches and Consoles
Commit
Pull Requests
Bookmarks
Structure
AWS Toolkit
Database
Gradle
Device Manager
Notifications
Android Emulator

32 static public void main(String[] argv) {
33     var test = new ex45();
34
35     // Test the makeRegex() method.
36     test.testRegexList();
37
38     // Create a List of String objects containing the complete
39     // works of Shakespeare.
40     List<String> bardWorks = BardDataFactory
41         .getInput( filename: $SHAKESPEARE_DATA_FILE,
42                 // Split input into "works".
43                 "@" );
44
45     // Search the works of Shakespeare for a certain word/phrase.
46     test.processBardWorks( bardWorks: Objects.requireNonNull( obj: bardWorks),
47                           word: "lord" );
48 }
```

Git Run CodeWhisperer Reference Log Logcat Profiler Dependencies TODO Problems Terminal Services App Inspection
AWS: No credentials selected CodeWhisperer 31:8 CRLF UTF-8 4 spaces master

See [Java8/ex45/src/main/java/ex45.java](https://github.com/stephenwalton/ex45/blob/main/src/main/java/ex45.java)

End of Overview of the ex45 Case Study: the Driver Class & Runtime Behavior