

The ex45 Case Study:

ex45 Driver Class & Runtime Behavior

Douglas C. Schmidt

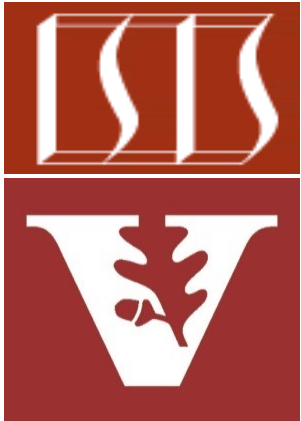
d.schmidt@vanderbilt.edu

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 program

ex45		
 	<code>sSHAKESPEARE_DATA_FILE</code>	String
 	<code>findMatch(String, String)</code>	boolean
 	<code>main(String[])</code>	void
 	<code>processBardWorks(List<String>, String)</code>	void
 	<code>showRegexMatches(List<String>, Pattern)</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/douglas-craig-schmidt/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 program
 - 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

ex45		
f	sSHAKESPEARE_DATA_FILE	String
m	findMatch(String, String)	boolean
m	main(String[])	void
m	processBardWorks(List<String>, String)	void
m	showRegexMatches(List<String>, Pattern)	void
m	testRegexList()	void

See [Java8/ex45/src/main/java/ex45.java](https://github.com/azul-systems/Java8/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

ex45		
f	sSHAKESPEARE_DATA_FILE	String
m	findMatch(String, String)	boolean
m	main(String[])	void
m	processBardWorks(List<String>, String)	void
m	showRegexMatches(List<String>, Pattern)	void
m	testRegexList()	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 & showRegexMatches() to display matching portions

ex45		
f	sSHAKESPEARE_DATA_FILE	String
m	findMatch(String, String)	boolean
m	main(String[])	void
m	processBardWorks(List<String>, String)	void
m	showRegexMatches(List<String>, Pattern)	void
m	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		
f	🔒	<code>sSHAKESPEARE_DATA_FILE</code> String
m	🔒	<code>findMatch</code> (String, String) boolean
m	🔓	<code>main</code> (String[]) void
m	🔒	<code>processBardWorks</code> (List<String>, String) void
m	🔒	<code>showRegexMatches</code> (List<String>, Pattern) void
m	🔒	<code>testRegexList</code> () void

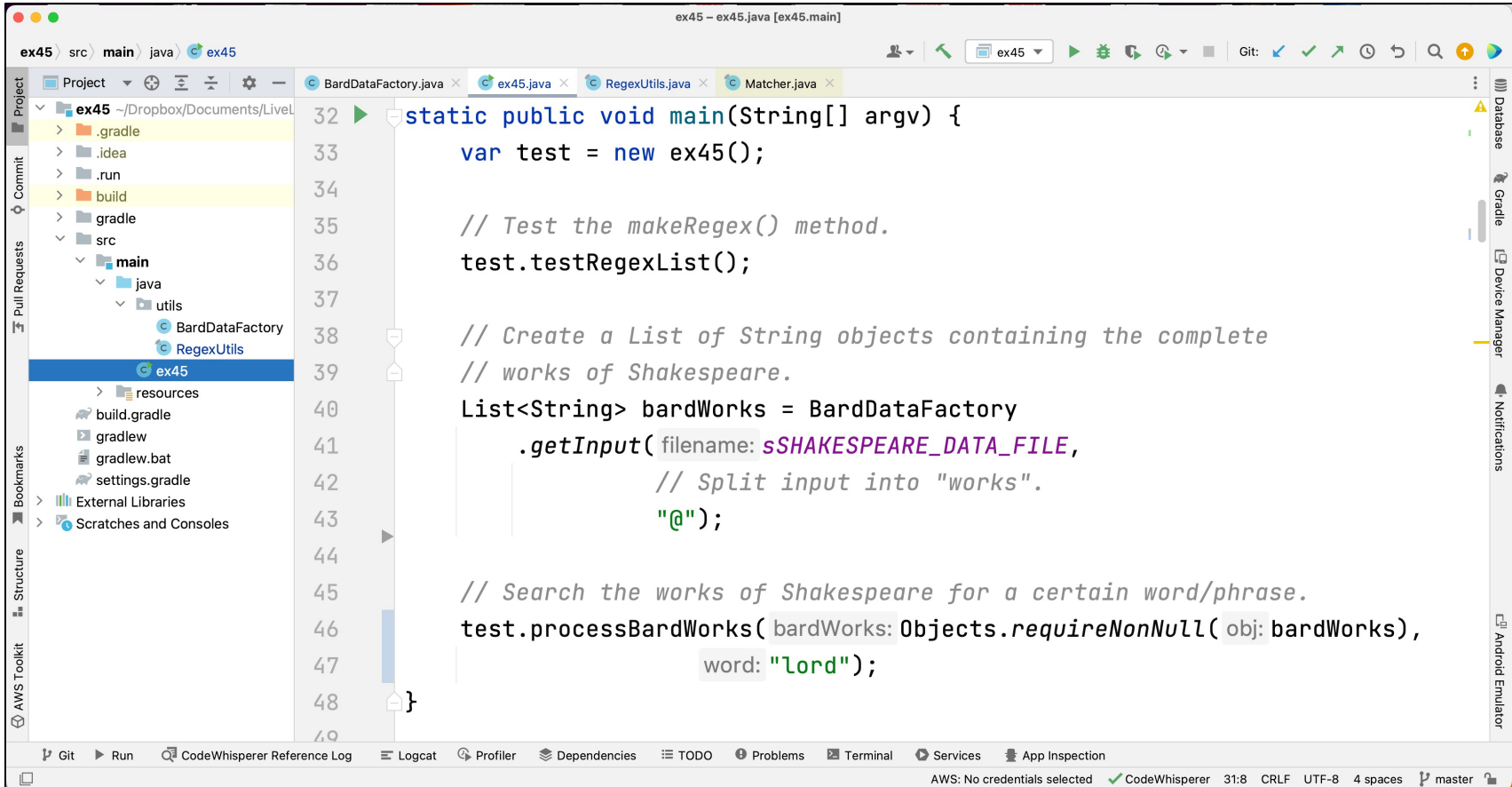
Methods in the ex45 Class

showRegexMatches ()

- Receives a list containing the works of Shakespeare & a compiled regular expression pattern
- Identifies portions of text in the works that match the pattern
- Collects & prints these matches organized by the title of each work

ex45		
f	sSHAKESPEARE_DATA_FILE	String
m	findMatch(String, String)	boolean
m	main(String[])	void
m	processBardWorks(List<String>, String)	void
m	showRegexMatches(List<String>, Pattern)	void
m	testRegexList()	void

Methods in the ex45 Class



The screenshot shows an IDE window titled "ex45 - ex45.java [ex45.main]". The left sidebar displays a project structure for "ex45" with folders for ".gradle", ".idea", ".run", "build", "gradle", "src", "main", "resources", and "utils". The "main" folder contains "BardDataFactory", "RegexUtils", and "ex45". The "utils" folder contains "BardDataFactory" and "RegexUtils". The main editor area shows the code for the "ex45.java" file, which is a simple Java class with a "main" method. The code is as follows:

```
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 }
```

The IDE interface includes a top toolbar with icons for file operations, a bottom toolbar with icons for Git, Run, CodeWhisperer Reference Log, Logcat, Profiler, Dependencies, TODO, Problems, Terminal, Services, and App Inspection, and a right sidebar with icons for Database, Gradle, Device Manager, Notifications, and Android Emulator. The status bar at the bottom indicates "AWS: No credentials selected", "CodeWhisperer", "31:8", "CRLF", "UTF-8", "4 spaces", and "master".

See [Java8/ex45/src/main/java/ex45.java](https://github.com/Ex45/ex45/blob/main/src/main/java/ex45.java)

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