Applying Java Functional Programming Features to the ThreadJoinTest Case Study

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

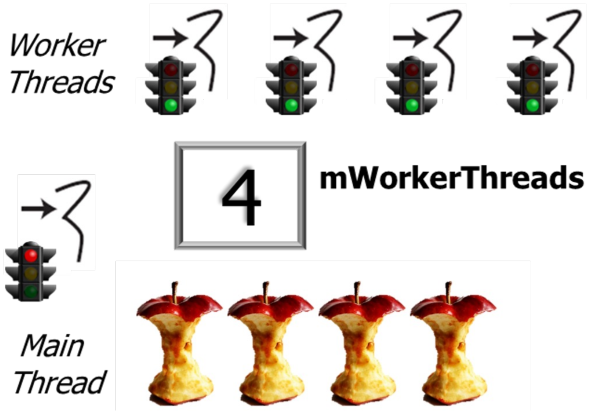
Professor of Computer Science

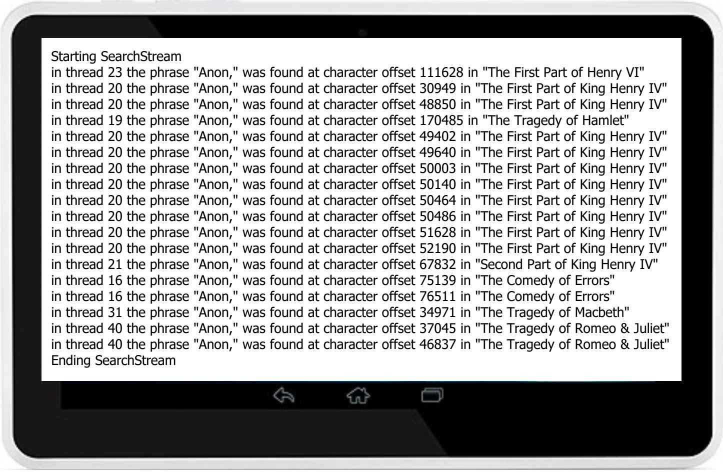
Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand how Java functional programming features are applied in multi-threaded program



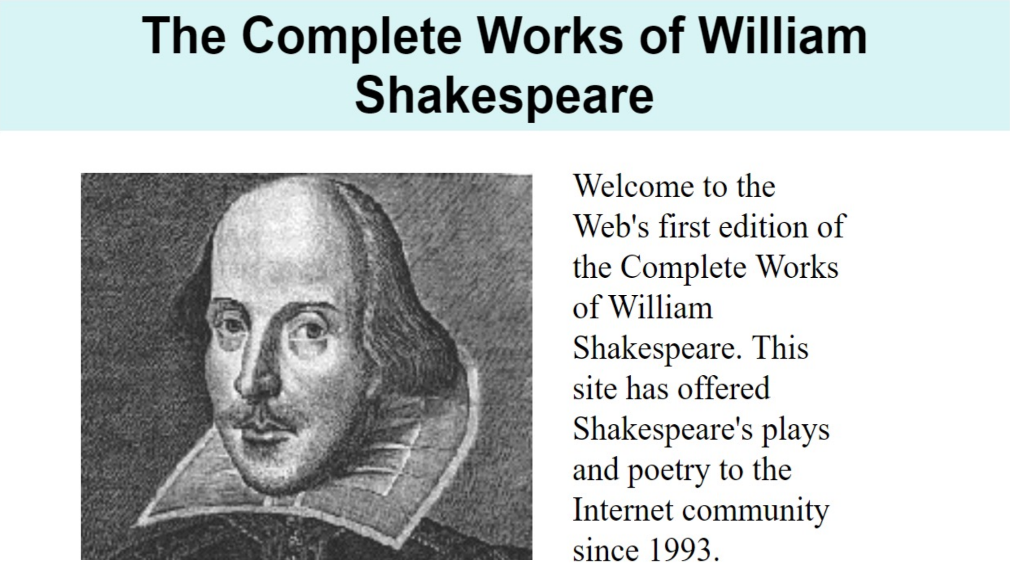


```
Starting SearchStream
in thread 23 the phrase "Anon," was found at character offset 111628 in "The First Part of Henry VI"
in thread 20 the phrase "Anon," was found at character offset 30949 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 48850 in "The First Part of King Henry IV"
in thread 19 the phrase "Anon," was found at character offset 170485 in "The Tragedy of Hamlet"
in thread 20 the phrase "Anon," was found at character offset 49402 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 49640 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50003 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50140 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50464 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50486 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 51628 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 52190 in "The First Part of King Henry IV"
in thread 21 the phrase "Anon," was found at character offset 67832 in "Second Part of King Henry IV"
in thread 16 the phrase "Anon," was found at character offset 75139 in "The Comedy of Errors"
in thread 16 the phrase "Anon," was found at character offset 76511 in "The Comedy of Errors"
in thread 31 the phrase "Anon," was found at character offset 34971 in "The Tragedy of Macbeth"
in thread 40 the phrase "Anon," was found at character offset 37045 in "The Tragedy of Romeo & Juliet"
in thread 40 the phrase "Anon," was found at character offset 46837 in "The Tragedy of Romeo & Juliet"
Ending SearchStream
```

See github.com/douglascraigschmidt/ModernJava/tree/main/CS/ThreadJoinTest

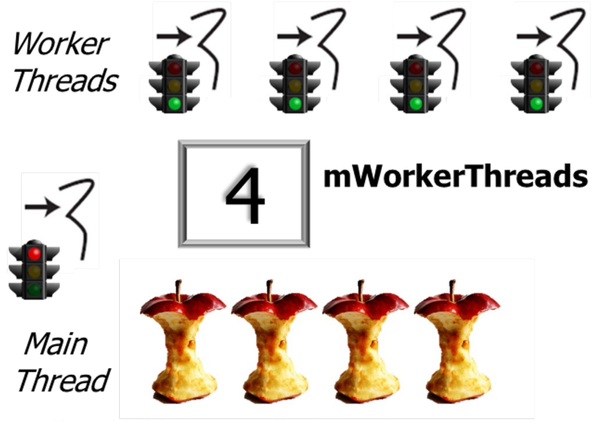
Learning Objectives in this Part of the Lesson

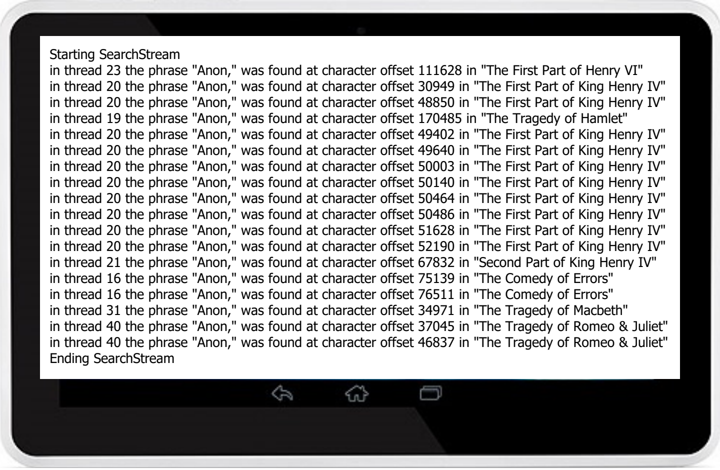
- Understand how Java functional programming features are applied in multi-threaded program
  - This program searches for a list of phrases in the complete works of William Shakespeare



The Complete Works of William Shakespeare

Welcome to the Web's first edition of the Complete Works of William Shakespeare. This site has offered Shakespeare's plays and poetry to the Internet community since 1993.



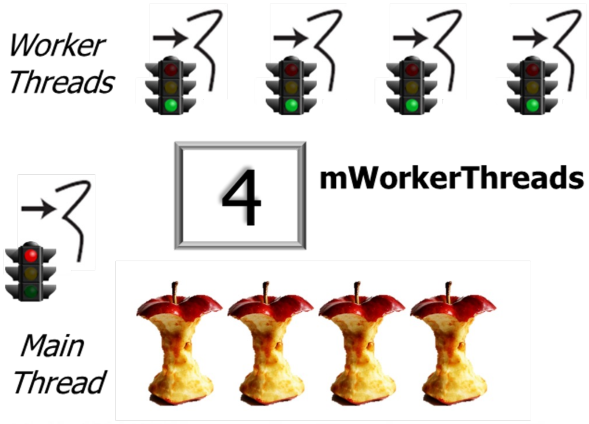


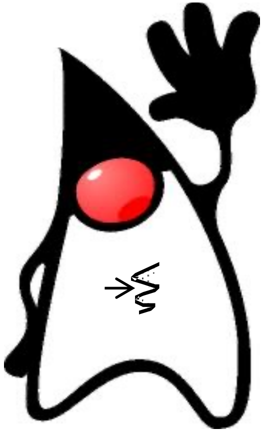
```
Starting SearchStream
in thread 23 the phrase "Anon," was found at character offset 111628 in "The First Part of Henry VI"
in thread 20 the phrase "Anon," was found at character offset 30949 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 48850 in "The First Part of King Henry IV"
in thread 19 the phrase "Anon," was found at character offset 170485 in "The Tragedy of Hamlet"
in thread 20 the phrase "Anon," was found at character offset 49402 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 49640 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50003 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50140 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50464 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50486 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 51628 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 52190 in "The First Part of King Henry IV"
in thread 21 the phrase "Anon," was found at character offset 67832 in "Second Part of King Henry IV"
in thread 16 the phrase "Anon," was found at character offset 75139 in "The Comedy of Errors"
in thread 16 the phrase "Anon," was found at character offset 76511 in "The Comedy of Errors"
in thread 31 the phrase "Anon," was found at character offset 34971 in "The Tragedy of Macbeth"
in thread 40 the phrase "Anon," was found at character offset 37045 in "The Tragedy of Romeo & Juliet"
in thread 40 the phrase "Anon," was found at character offset 46837 in "The Tragedy of Romeo & Juliet"
Ending SearchStream
```

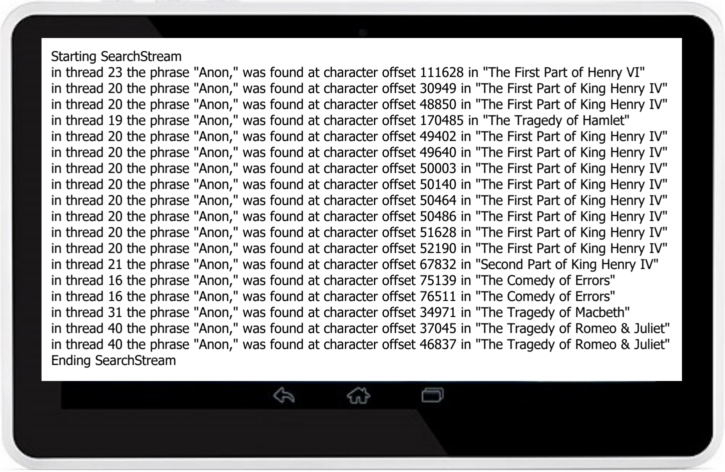
See shakespeare.mit.edu

Learning Objectives in this Part of the Lesson

- Understand how Java functional programming features are applied in multi-threaded program
  - This program searches for a list of phrases in the complete works of William Shakespeare
  - It also demonstrates the use of Java Thread objects







```
Starting SearchStream
in thread 23 the phrase "Anon," was found at character offset 111628 in "The First Part of Henry VI"
in thread 20 the phrase "Anon," was found at character offset 30949 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 48850 in "The First Part of King Henry IV"
in thread 19 the phrase "Anon," was found at character offset 170485 in "The Tragedy of Hamlet"
in thread 20 the phrase "Anon," was found at character offset 49402 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 49640 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50003 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50140 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50464 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50486 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 51628 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 52190 in "The First Part of King Henry IV"
in thread 21 the phrase "Anon," was found at character offset 67832 in "Second Part of King Henry IV"
in thread 16 the phrase "Anon," was found at character offset 75139 in "The Comedy of Errors"
in thread 16 the phrase "Anon," was found at character offset 76511 in "The Comedy of Errors"
in thread 31 the phrase "Anon," was found at character offset 34971 in "The Tragedy of Macbeth"
in thread 40 the phrase "Anon," was found at character offset 37045 in "The Tragedy of Romeo & Juliet"
in thread 40 the phrase "Anon," was found at character offset 46837 in "The Tragedy of Romeo & Juliet"
Ending SearchStream
```

See docs.oracle.com/javase/8/docs/api/java/lang/Thread.html

Overview of the ThreadJoin Test Case Study Program

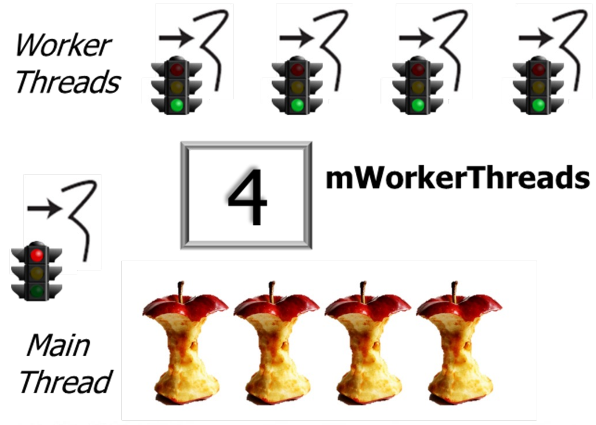
Overview of the ThreadJoinTest Case Study Program

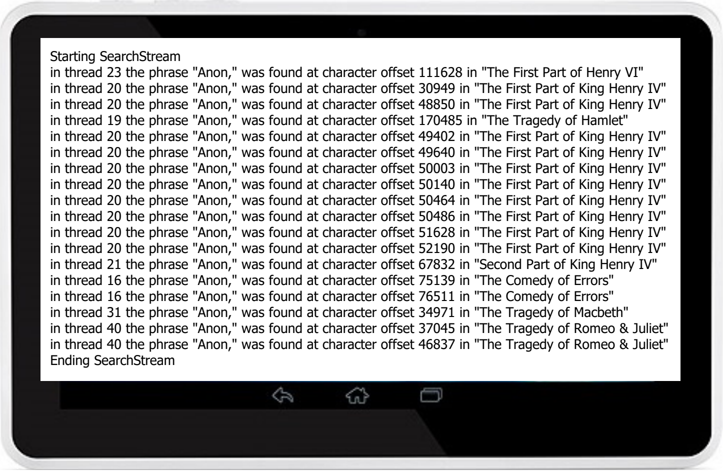
- Use Java functional programming features to create, start, & join Thread objects that concurrently search for phrases in works of Shakespeare

```
var workerThreads = ThreadUtils
   .makeThreads(mInputList,
                this::processInput);

workerThreads
   .forEach(Thread::start);

workerThreads
   .forEach(rethrowConsumer
              (Thread::join));
```



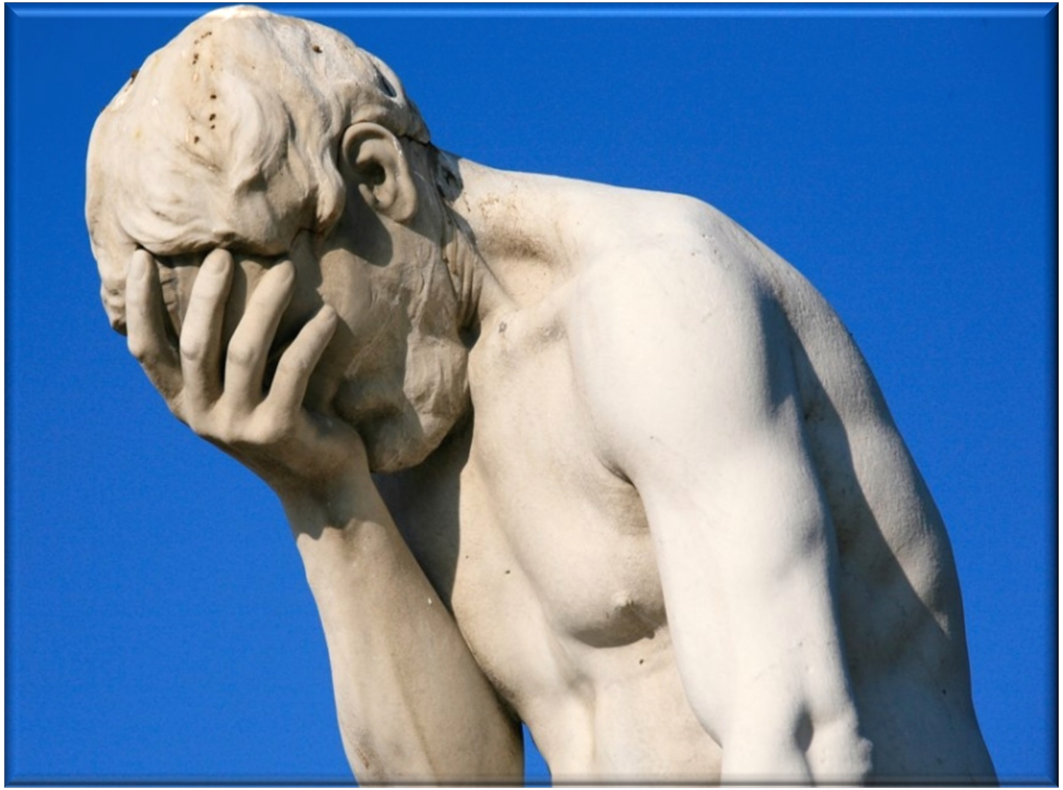


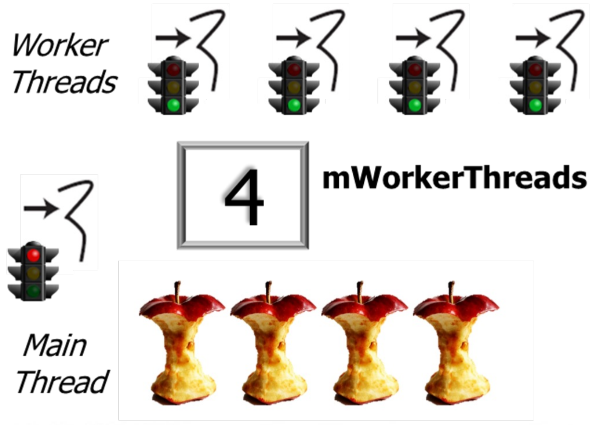
```
Starting SearchStream
in thread 23 the phrase "Anon," was found at character offset 111628 in "The First Part of Henry VI"
in thread 20 the phrase "Anon," was found at character offset 30949 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 48850 in "The First Part of King Henry IV"
in thread 19 the phrase "Anon," was found at character offset 170485 in "The Tragedy of Hamlet"
in thread 20 the phrase "Anon," was found at character offset 49402 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 49640 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50003 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50140 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50464 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50486 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 51628 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 52190 in "The First Part of King Henry IV"
in thread 21 the phrase "Anon," was found at character offset 67832 in "Second Part of King Henry IV"
in thread 16 the phrase "Anon," was found at character offset 75139 in "The Comedy of Errors"
in thread 16 the phrase "Anon," was found at character offset 76511 in "The Comedy of Errors"
in thread 31 the phrase "Anon," was found at character offset 34971 in "The Tragedy of Macbeth"
in thread 40 the phrase "Anon," was found at character offset 37045 in "The Tragedy of Romeo & Juliet"
in thread 40 the phrase "Anon," was found at character offset 46837 in "The Tragedy of Romeo & Juliet"
Ending SearchStream
```

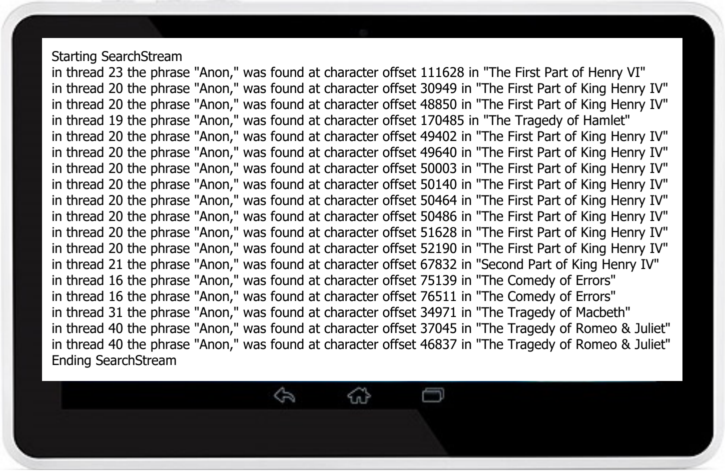
See github.com/douglascraigschmidt/ModernJava/tree/main/CS/ThreadJoinTest

Overview of the ThreadJoinTest Case Study Program

- This program is "embarrassingly parallel"





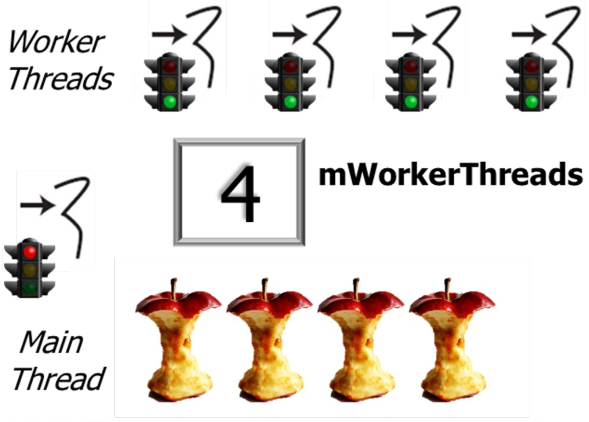


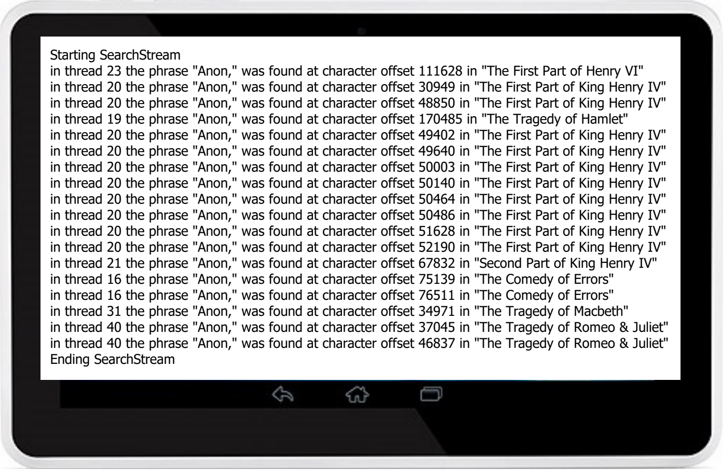
```
Starting SearchStream
in thread 23 the phrase "Anon," was found at character offset 111628 in "The First Part of Henry VI"
in thread 20 the phrase "Anon," was found at character offset 30949 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 48850 in "The First Part of King Henry IV"
in thread 19 the phrase "Anon," was found at character offset 170485 in "The Tragedy of Hamlet"
in thread 20 the phrase "Anon," was found at character offset 49402 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 49640 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50003 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50140 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50464 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50486 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 51628 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 52190 in "The First Part of King Henry IV"
in thread 21 the phrase "Anon," was found at character offset 67832 in "Second Part of King Henry IV"
in thread 16 the phrase "Anon," was found at character offset 75139 in "The Comedy of Errors"
in thread 16 the phrase "Anon," was found at character offset 76511 in "The Comedy of Errors"
in thread 31 the phrase "Anon," was found at character offset 34971 in "The Tragedy of Macbeth"
in thread 40 the phrase "Anon," was found at character offset 37045 in "The Tragedy of Romeo & Juliet"
in thread 40 the phrase "Anon," was found at character offset 46837 in "The Tragedy of Romeo & Juliet"
Ending SearchStream
```

See en.wikipedia.org/wiki/Embarrassingly_parallel

Overview of the ThreadJoinTest Case Study Program

- This program is "embarrassingly parallel"
  - i.e., there are no data dependencies between worker threads





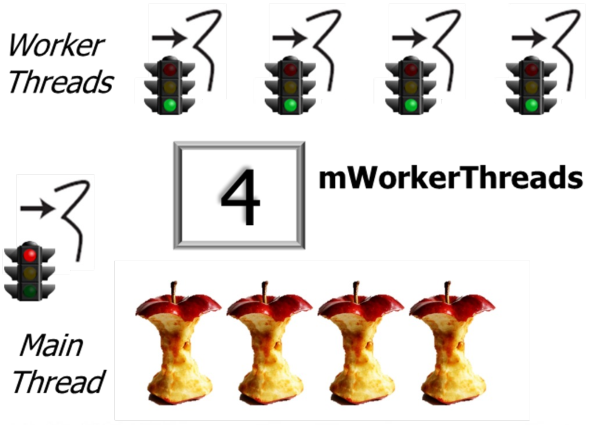
```
Starting SearchStream
in thread 23 the phrase "Anon," was found at character offset 111628 in "The First Part of Henry VI"
in thread 20 the phrase "Anon," was found at character offset 30949 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 48850 in "The First Part of King Henry IV"
in thread 19 the phrase "Anon," was found at character offset 170485 in "The Tragedy of Hamlet"
in thread 20 the phrase "Anon," was found at character offset 49402 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 49640 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50003 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50140 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50464 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50486 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 51628 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 52190 in "The First Part of King Henry IV"
in thread 21 the phrase "Anon," was found at character offset 67832 in "Second Part of King Henry IV"
in thread 16 the phrase "Anon," was found at character offset 75139 in "The Comedy of Errors"
in thread 16 the phrase "Anon," was found at character offset 76511 in "The Comedy of Errors"
in thread 31 the phrase "Anon," was found at character offset 34971 in "The Tragedy of Macbeth"
in thread 40 the phrase "Anon," was found at character offset 37045 in "The Tragedy of Romeo & Juliet"
in thread 40 the phrase "Anon," was found at character offset 46837 in "The Tragedy of Romeo & Juliet"
Ending SearchStream
```

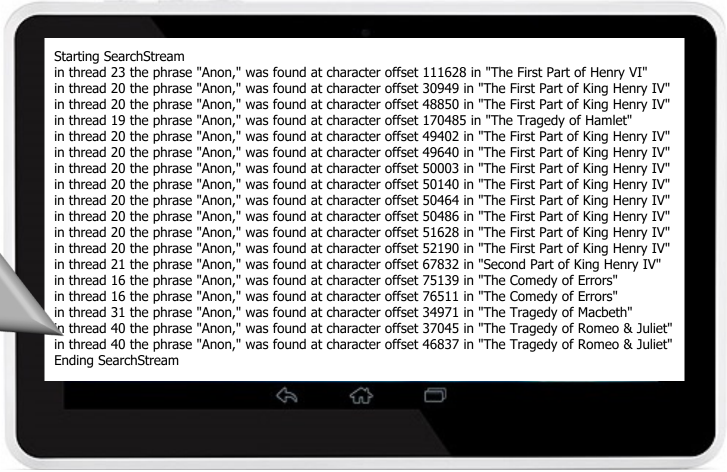
A laundromat is an apt metaphor for "embarrassingly parallel" processing!

Overview of the ThreadJoinTest Case Study Program

- This program is "embarrassingly parallel"
  - i.e., there are no data dependencies between worker threads

```
Starting SearchStream
in thread 23 the phrase "Anon," was found at character offset 111628 in "The First Part of Henry VI"
in thread 20 the phrase "Anon," was found at character offset 30949 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 48850 in "The First Part of King Henry IV"
in thread 19 the phrase "Anon," was found at character offset 170485 in "The Tragedy of Hamlet"
in thread 20 the phrase "Anon," was found at character offset 49402 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 49640 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50003 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50140 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50464 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50486 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 51628 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 52190 in "The First Part of King Henry IV"
in thread 21 the phrase "Anon," was found at character offset 67832 in "Second Part of King Henry IV"
in thread 16 the phrase "Anon," was found at character offset 75139 in "The Comedy of Errors"
in thread 16 the phrase "Anon," was found at character offset 76511 in "The Comedy of Errors"
in thread 31 the phrase "Anon," was found at character offset 34971 in "The Tragedy of Macbeth"
in thread 40 the phrase "Anon," was found at character offset 37045 in "The Tragedy of Romeo & Juliet"
in thread 40 the phrase "Anon," was found at character offset 46837 in "The Tragedy of Romeo & Juliet"
Ending SearchStream
```





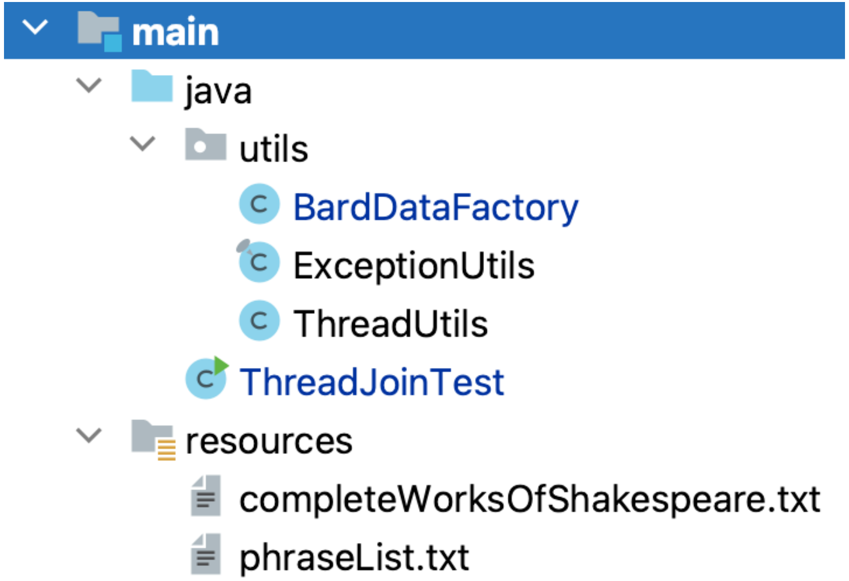
```
Starting SearchStream
in thread 23 the phrase "Anon," was found at character offset 111628 in "The First Part of Henry VI"
in thread 20 the phrase "Anon," was found at character offset 30949 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 48850 in "The First Part of King Henry IV"
in thread 19 the phrase "Anon," was found at character offset 170485 in "The Tragedy of Hamlet"
in thread 20 the phrase "Anon," was found at character offset 49402 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 49640 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50003 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50140 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50464 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 50486 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 51628 in "The First Part of King Henry IV"
in thread 20 the phrase "Anon," was found at character offset 52190 in "The First Part of King Henry IV"
in thread 21 the phrase "Anon," was found at character offset 67832 in "Second Part of King Henry IV"
in thread 16 the phrase "Anon," was found at character offset 75139 in "The Comedy of Errors"
in thread 16 the phrase "Anon," was found at character offset 76511 in "The Comedy of Errors"
in thread 31 the phrase "Anon," was found at character offset 34971 in "The Tragedy of Macbeth"
in thread 40 the phrase "Anon," was found at character offset 37045 in "The Tragedy of Romeo & Juliet"
in thread 40 the phrase "Anon," was found at character offset 46837 in "The Tragedy of Romeo & Juliet"
Ending SearchStream
```

Each work of Shakespeare is processed concurrently in its own Java Thread object

Structure & Functionality of the ThreadJoinTest Program

Structure & Functionality of the ThreadJoinTest Program

- The ThreadJoinTest project source code is organized into several files & packages

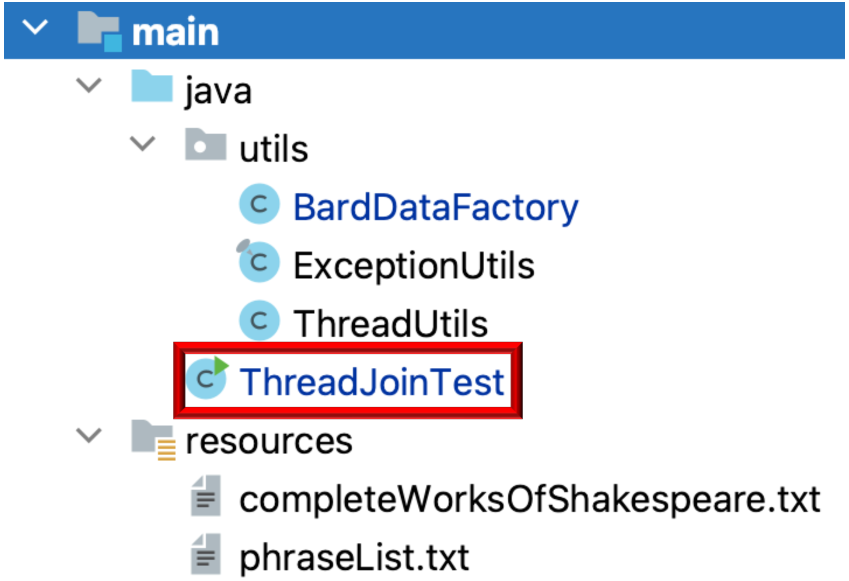


```
main
  java
    utils
      BardDataFactory
      ExceptionUtils
      ThreadUtils
    ThreadJoinTest
  resources
    completeWorksOfShakespeare.txt
    phraseList.txt
```

See github.com/douglascraigschmidt/ModernJava/tree/main/CS/ThreadJoinTest

Structure & Functionality of the ThreadJoinTest Program

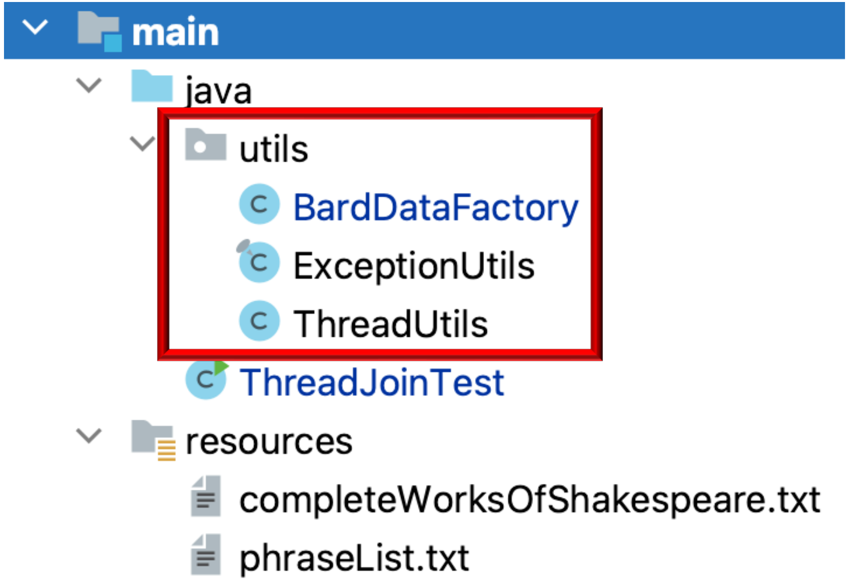
- The ThreadJoinTest project source code is organized into several files & packages
  - ThreadJoinTest
    - The main test driver program that concurrently searches for phrases in the complete works of William Shakespeare



- main
  - java
    - utils
      - BardDataFactory
      - ExceptionUtils
      - ThreadUtils
    - ThreadJoinTest
  - resources
    - completeWorksOfShakespeare.txt
    - phraseList.txt

Structure & Functionality of the ThreadJoinTest Program

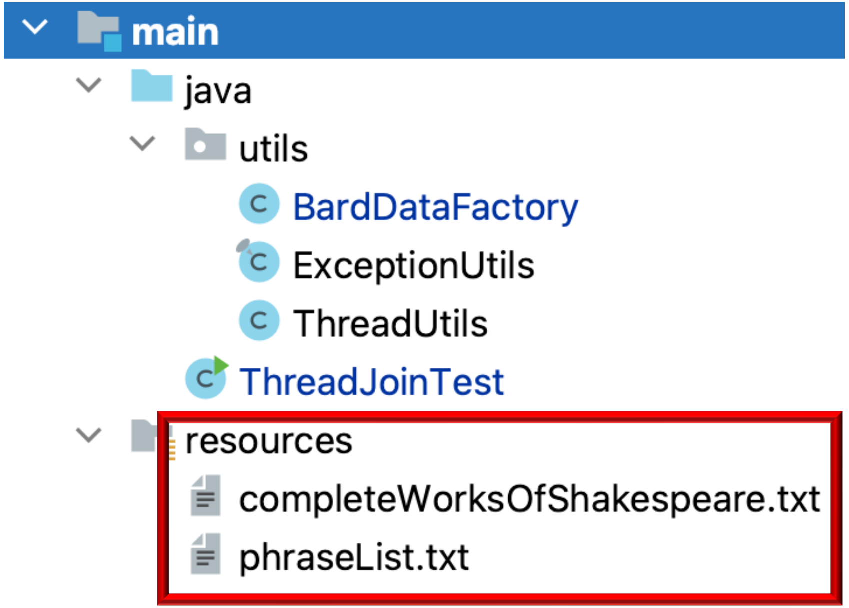
- The ThreadJoinTest project source code is organized into several files & packages
  - ThreadJoinTest
  - utils
    - Contains general-purpose reusable classes



```
main
  java
    utils
      BardDataFactory
      ExceptionUtils
      ThreadUtils
    ThreadJoinTest
  resources
    completeWorksOfShakespeare.txt
    phraseList.txt
```

Structure & Functionality of the ThreadJoinTest Program

- The ThreadJoinTest project source code is organized into several files & packages
  - ThreadJoinTest
  - utils
  - resources
    - Contains the complete works of Shakespeare & the list of Bard phrases



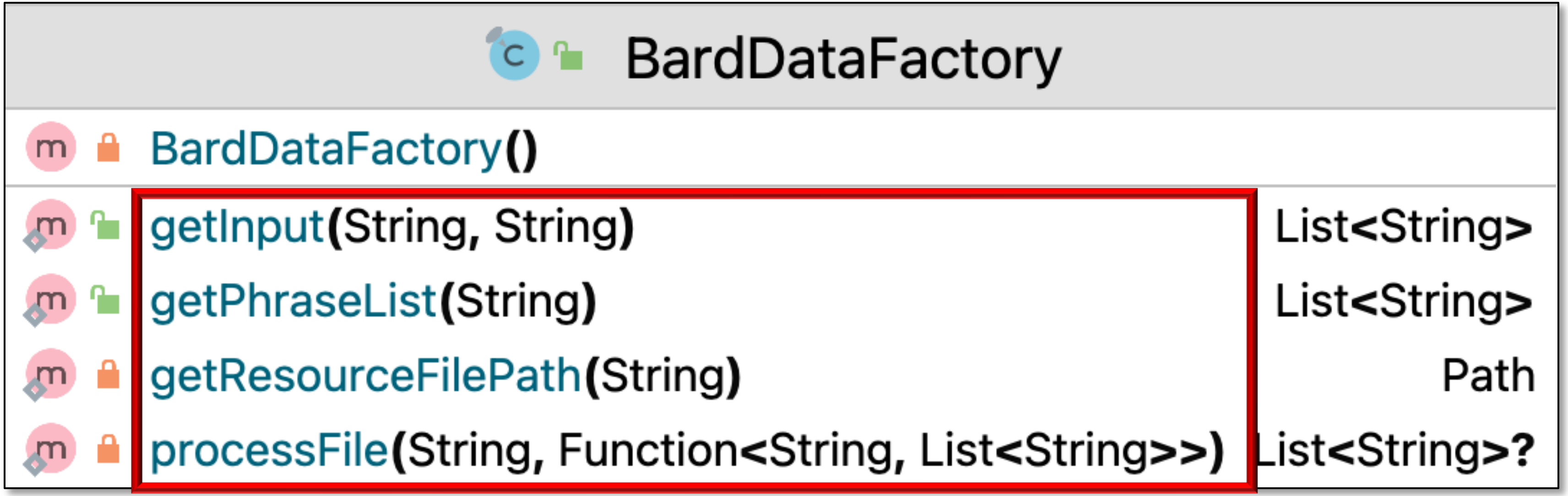
```
main
  java
    utils
      BardDataFactory
      ExceptionUtils
      ThreadUtils
    ThreadJoinTest
  resources
    completeWorksOfShakespeare.txt
    phraseList.txt
```

See christirapt.wordpress.com/2017/11/24/why-is-shakespeare-called-the-bard

Obtaining the ThreadJoinTest Program’s Input Data

Obtaining the ThreadJoinTest Program’s Input Data

- The BardDataFactory utility class applies functional programming features to read the contents of text files & return this content as Lists of String objects



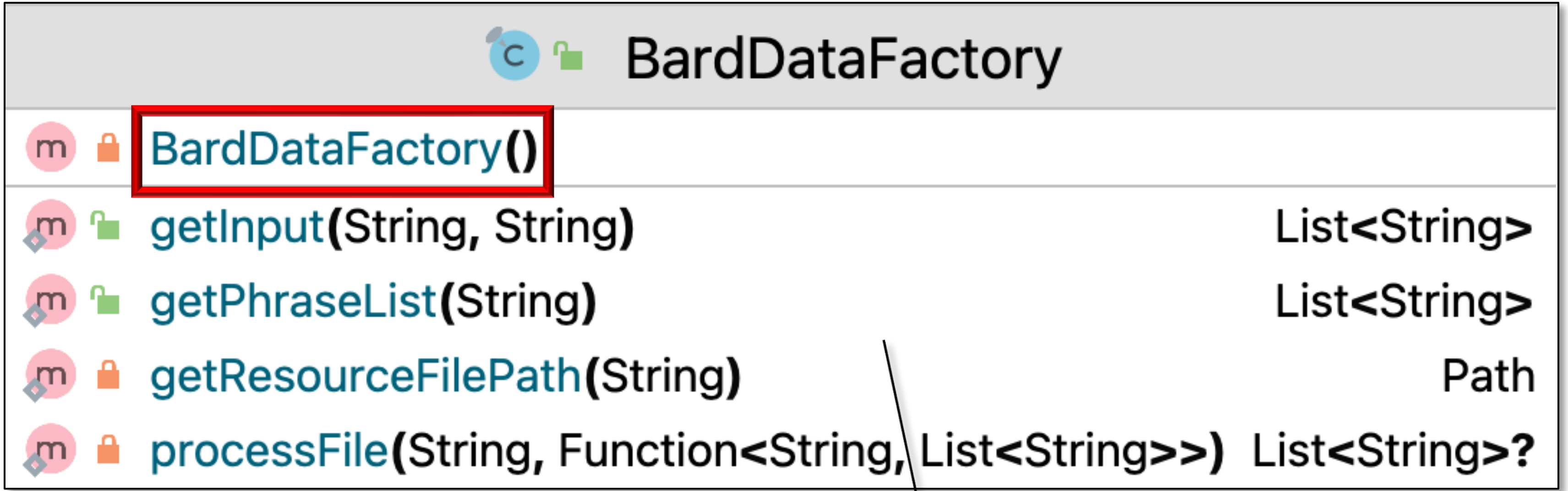
BardDataFactory

BardDataFactory()	
getInput(String, String)	List<String>
getPhraseList(String)	List<String>
getResourceFilePath(String)	Path
processFile(String, Function<String, List<String>>)	List<String>?

See CS/ThreadJoinTest/src/main/java/utils/BardDataFactory.java

Obtaining the ThreadJoinTest Program’s Input Data

- The BardDataFactory utility class applies functional programming features to read the contents of text files & return this content as Lists of String objects



BardDataFactory

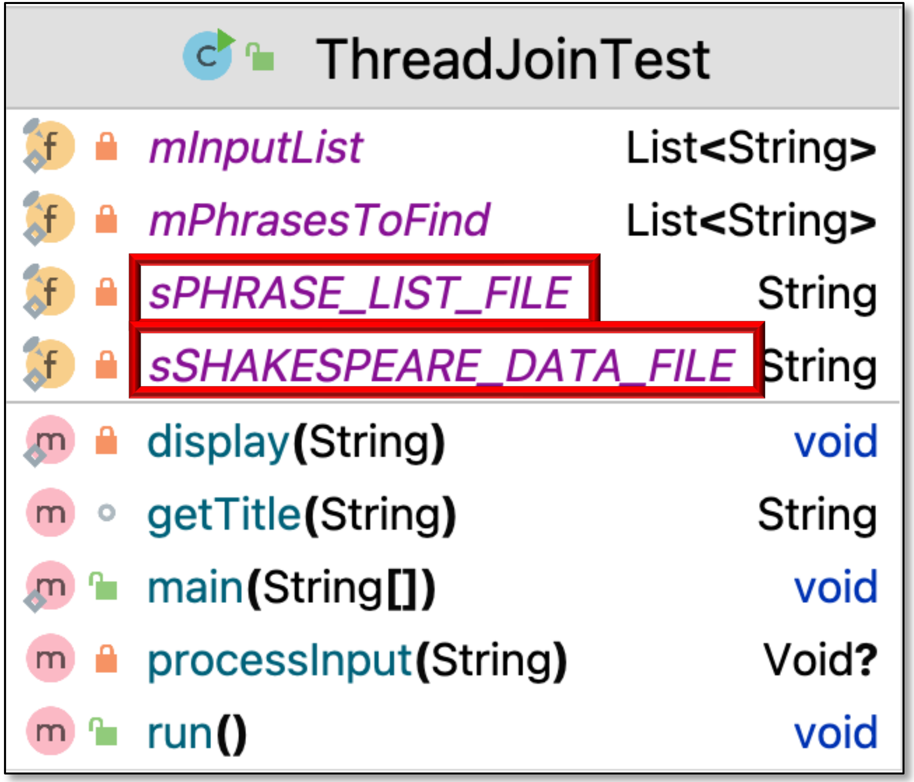
BardDataFactory()	
getInput(String, String)	List<String>
getPhraseList(String)	List<String>
getResourceFilePath(String)	Path
processFile(String, Function<String, List<String>>)	List<String>?

A utility class in Java is defined as final & has only static methods, no non-static state, & a private constructor

See www.quora.com/What-is-the-best-way-to-write-utility-classes-in-Java/answer/Jon-Harley

Obtaining the ThreadJoinTest Program’s Input Data

- The BardDataFactory reads the works of Shakespeare & the Bard phrases to search for from two text files



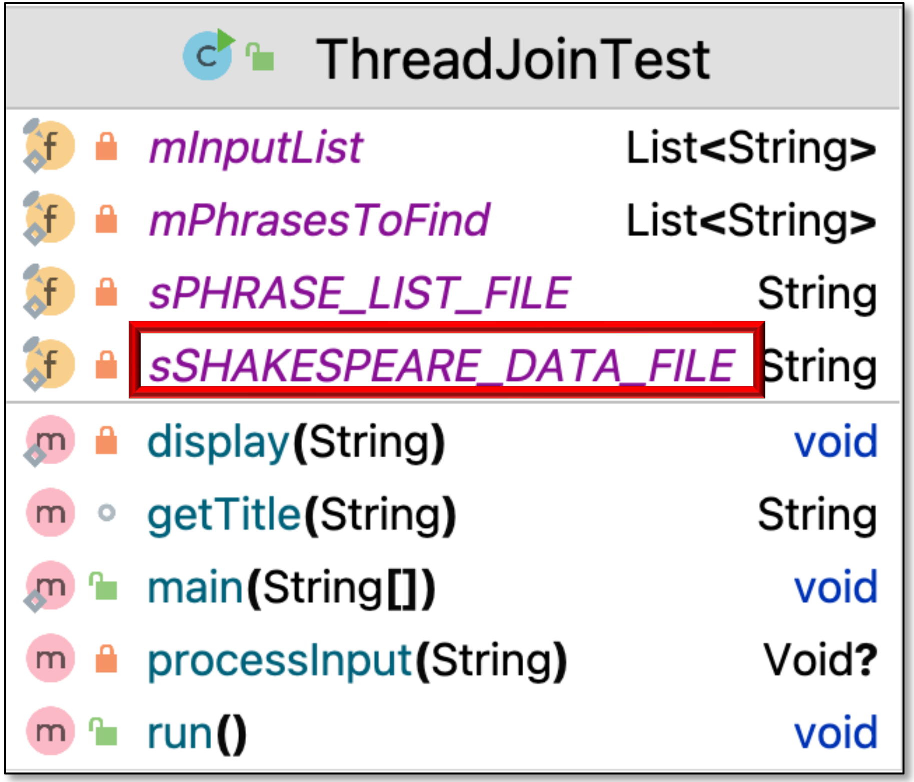
ThreadJoinTest

mInputList	List<String>
mPhrasesToFind	List<String>
sPHRASE_LIST_FILE	String
sSHAKESPEARE_DATA_FILE	String
display(String)	void
getTitle(String)	String
main(String[])	void
processInput(String)	Void?
run()	void

Obtaining the ThreadJoinTest Program's Input Data

- The BardDataFactory reads the works of Shakespeare & the Bard phrases to search for from two text files

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



ThreadJoinTest

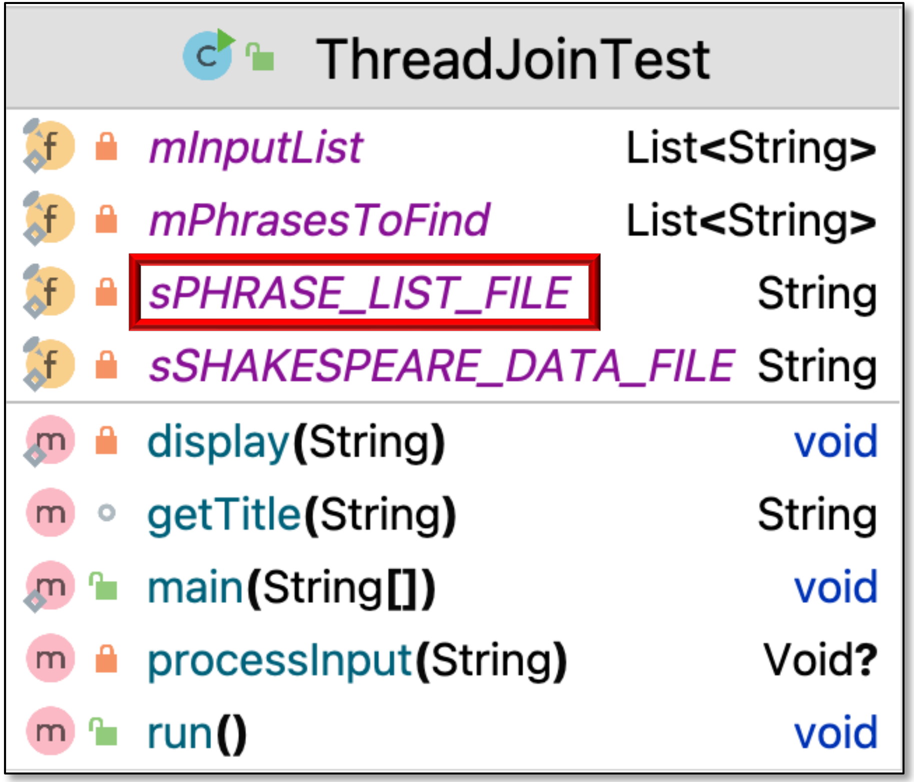
mInputList	List<String>
mPhrasesToFind	List<String>
sPHRASE_LIST_FILE	String
sSHAKESPEARE_DATA_FILE	String
display(String)	void
getTitle(String)	String
main(String[])	void
processInput(String)	Void?
run()	void

Each work begins with '@', which is matched by a Java regular expression

Obtaining the ThreadJoinTest Program's Input Data

- The BardDataFactory reads the works of Shakespeare & the Bard phrases to search for from two text files

```
List<String> mPhrasesToFind =
   BardDataFactory.getPhraseList
      (sPHRASE_LIST_FILE);
...
Beware the Ides of March
Brevity is the soul of wit
All that glisters is not gold
Sit you down, father; rest you
my kingdom for a horse!
Off with his head!
Neither a borrower nor a lender be ...
```



ThreadJoinTest

mInputList	List<String>
mPhrasesToFind	List<String>
sPHRASE_LIST_FILE	String
sSHAKESPEARE_DATA_FILE	String
display(String)	void
getTitle(String)	String
main(String[])	void
processInput(String)	Void?
run()	void

Each phrase appears on a separate line

Obtaining the ThreadJoinTest Program's Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
   (String filename, Function<String, List<String>> fileProc) {
   var uri = getResourcePathFile(filename);


   String contents = new String(Files.readAllBytes(uri));


   var strings = fileProc.apply(contents);

   strings.removeIf(String::isEmpty);

   return strings;
}
```

See CS/ThreadJoinTest/src/main/java/utils/BardDataFactory.java

Obtaining the ThreadJoinTest Program’s Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
   (String filename, Function<String, List<String>> fileProc) {
   var uri = getResourcePathFile(filename);

   String contents = new String(Files.readAllBytes(uri));


   var strings = fileProc.apply(contents);

   strings.removeIf(String::isEmpty);

   return strings;
}
```

This filename identifies a File to process

Obtaining the ThreadJoinTest Program's Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
  (String filename, Function<String, List<String>> fileProc) {
  var uri = getResourcePathFile(filename);

  String contents = new String(Files.readAllBytes(uri));

  var strings = fileProc.apply(contents);

  strings.removeIf(String::isEmpty);

  return strings;
}
```

The Function that processes the file in some manner

This method applies a Java Function to factor out common code

Obtaining the ThreadJoinTest Program’s Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
   (String filename, Function<String, List<String>> fileProc) {
   var uri = getResourcePathFile(filename);
```

Convert filename into a Path URI

```
   String contents = new String(Files.readAllBytes(uri));


   var strings = fileProc.apply(contents);

   strings.removeIf(String::isEmpty);

   return strings;
}
```

Obtaining the ThreadJoinTest Program's Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
   (String filename, Function<String, List<String>> fileProc) {
   var uri = getResourcePathFile(filename);
```

```
static Path getResourceFilePath(String filename)
    throws URISyntaxException {
    return Paths
      .get(ClassLoader.getSystemResource(filename)
      .toURI());
}
```

```
   return strings;
}
```

This helper method gets a Path to the designated File in the resources folder

Obtaining the ThreadJoinTest Program's Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
   (String filename, Function<String, List<String>> fileProc) {
   var uri = getResourcePathFile(filename);

   String contents = new String(Files.readAllBytes(uri));


   var strings = fileProc.apply(contents);

   strings.removeIf(String::isEmpty);

   return strings;
}
```

Open the File & read its contents into a String

See docs.oracle.com/javase/8/docs/api/java/nio/file/Files.html#readAllBytes

Obtaining the ThreadJoinTest Program's Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
   (String filename, Function<String, List<String>> fileProc) {
   var uri = getResourcePathFile(filename);


   String contents = new String(Files.readAllBytes(uri));


   var strings = fileProc.apply(contents);

   strings.removeIf(String::isEmpty);

   return strings;
}
```

Apply the Function to process the File contents & return a List of String objects

See docs.oracle.com/javase/8/docs/api/java/util/function/Function.html#apply

Obtaining the ThreadJoinTest Program's Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
   (String filename, Function<String, List<String>> fileProc) {
   var uri = getResourcePathFile(filename);


   String contents = new String(Files.readAllBytes(uri));


   var strings = fileProc.apply(contents);

   strings.removeIf(String::isEmpty);

   return strings;
}
```

Use a Predicate method reference to remove empty String objects in the List

See docs.oracle.com/javase/8/docs/api/java/util/Collection.html#removeIf

Obtaining the ThreadJoinTest Program’s Input Data

- Process the contents of a File associated with filename using the Function

```
static List<String> processFile
  (String filename, Function<String, List<String>> fileProc) {
  var uri = getResourcePathFile(filename);


  String contents = new String(Files.readAllBytes(uri));


  var strings = fileProc.apply(contents);

  strings.removeIf(String::isEmpty);

  return strings;
}
```

Return the List of non-empty String objects

Obtaining the ThreadJoinTest Program’s Input Data

- Return contents associated w/filename as a List of non-empty String objects

```
static List<String> getInput(String filename, String splitter) {



    return processFile
      (filename,
       content -> Arrays.asList(content.split(splitter)));
}
```

Obtaining the ThreadJoinTest Program's Input Data

- Return contents associated w/filename as a List of non-empty String objects

```
static List<String> getInput(String filename, String splitter) {
```

Identifies the File containing the complete works of Shakespeare

```
  return processFile
    (filename,
     content -> Arrays.asList(content.split(splitter)));
}
```

Obtaining the ThreadJoinTest Program's Input Data

- Return contents associated w/filename as a List of non-empty String objects

```
static List<String> getInput(String filename, String splitter) {

  return processFile
    (filename,
     content -> Arrays.asList(content.split(splitter)));
}
```

This splitter is the '@' character that separates each work of Shakespeare

The '@' character is used as a regular expression to split the File into works

Obtaining the ThreadJoinTest Program’s Input Data

- Return contents associated w/filename as a List of non-empty String objects

```
static List<String> getInput(String filename, String splitter) {




   return processFile
     (filename,
      content -> Arrays.asList(content.split(splitter)));
}
```

Call processFile(), passing a Function that splits the File associated with 'filename' by the 'splitter’ regular expression & returns a List containing the complete works of Shakespeare

See docs.oracle.com/javase/8/docs/api/java/lang/String.html#split

Obtaining the ThreadJoinTest Program's Input Data

- Return phrases associated w/filename as a List of non-empty String objects

```
static List<String> getPhraseList(String filename) {




  return processFile
    (filename,
     content -> Arrays.asList(content.split("\\R")));
}
```

See CS/ThreadJoinTest/src/main/java/utils/BardDataFactory.java

Obtaining the ThreadJoinTest Program’s Input Data

- Return phrases associated w/filename as a List of non-empty String objects

```
static List<String> getPhraseList(String filename) {
```

Identifies the File containing the Bard phrases to search for

```
  return processFile
    (filename,
     content -> Arrays.asList(content.split("\\R")));
}
```

Obtaining the ThreadJoinTest Program's Input Data

- Return phrases associated w/filename as a List of non-empty String objects

```
static List<String> getPhraseList(String filename) {



  return processFile
    (filename,
     content -> Arrays.asList(content.split("\\R")));
}
```

Call processFile(), passing a Function that splits the File associated with 'filename' by the newline regular expression & returns a List containing Bard phrases to search for

Obtaining the ThreadJoinTest Program’s Input Data

- Return phrases associated w/filename as a List of non-empty String objects

```
static List<String> getPhraseList(String filename) {




  return processFile
    (filename,
     content -> Arrays.asList(content.split("\\R")));
}
```

"\\R" is a regular expression that matches any line separator (e.g., "\r", "\n", or "\r\n”)

See docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html

Applying BardDataFactory in the ThreadJoinTest Program

Applying BardDataFactory in the ThreadJoinTest Program

- ThreadJoinTest defines static fields that are initialized by BardDataFactory methods

```
public class ThreadJoinTest {
  static final List<String>
    mInputList = BardDataFactory
      .getInput
         (sSHAKESPEARE_DATA_FILE,
          "@");

  static final List<String>
    mPhrasesToFind = BardDataFactory
      .getPhraseList
         (sPHRASE_LIST_FILE);
 ...
}
```

See CS/ThreadJoinTest/src/main/java/ThreadJoinTest.java

Applying BardDataFactory in the ThreadJoinTest Program

- ThreadJoinTest defines static fields that are initialized by BardDataFactory methods

```
public class ThreadJoinTest {
  static final List<String>
    mInputList = BardDataFactory
      .getInput
         (sSHAKESPEARE_DATA_FILE,
          "@");

  static final List<String>
    mPhrasesToFind = BardDataFactory
      .getPhraseList
         (sPHRASE_LIST_FILE);
 ...
}
```

A List of the complete works of Shakespeare

Applying BardDataFactory in the ThreadJoinTest Program

- ThreadJoinTest defines static fields that are initialized by BardDataFactory methods

```
public class ThreadJoinTest {
  static final List<String>
    mInputList = BardDataFactory
      .getInput
         (sSHAKESPEARE_DATA_FILE,
          "@");

  static final List<String>
    mPhrasesToFind = BardDataFactory
      .getPhraseList
         (sPHRASE_LIST_FILE);
 ...
}
```

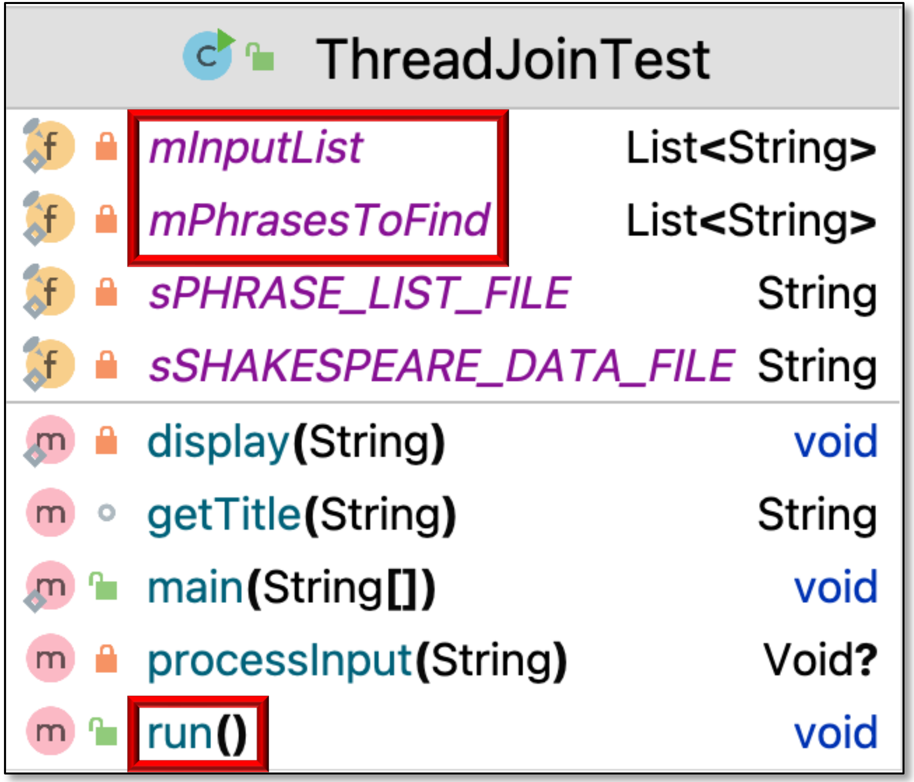
A List of the Bard phrases to search for

Applying BardDataFactory in the ThreadJoinTest Program

- The main program creates & runs an instance of ThreadJoinTest that uses these Lists to guide its processing

```
public void main(String[] a) {
  new ThreadJoinTest().run();
}
```

Create/run an object that searches for all Bard phrases concurrently



ThreadJoinTest

mInputList	List<String>
mPhrasesToFind	List<String>
sPHRASE_LIST_FILE	String
sSHAKESPEARE_DATA_FILE	String
display(String)	void
getTitle(String)	String
main(String[])	void
processInput(String)	Void?
run()	void

See CS/ThreadJoinTest/src/main/java/ThreadJoinTest.java

End of Applying Java Functional Programming Features to the ThreadJoin Test Case Study