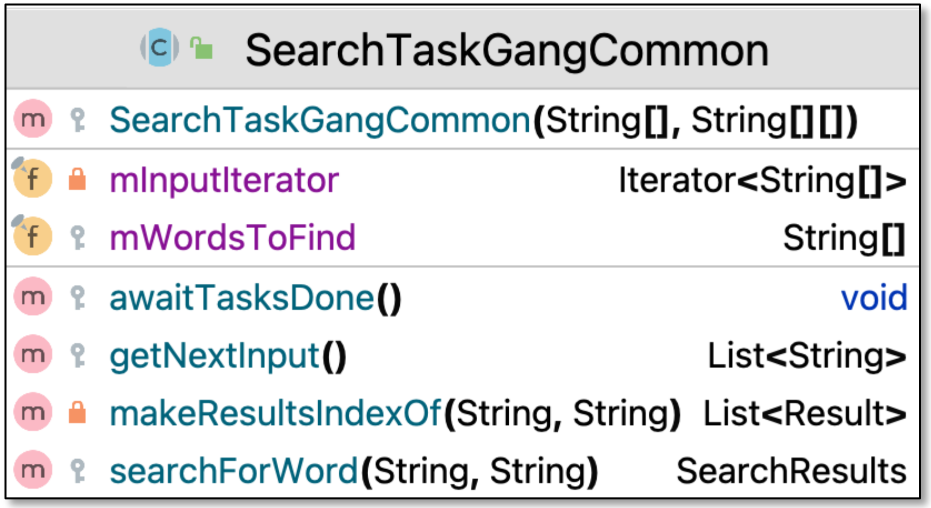
Overview of the Search TaskGangCommon Class

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

Institute for Software
Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand the SearchTaskGang case study
- Recognize the methods that are defined by the TaskGang framework
- Know the subclasses that extends TaskGang (directly or indirectly)
  - SearchTaskGangCommon



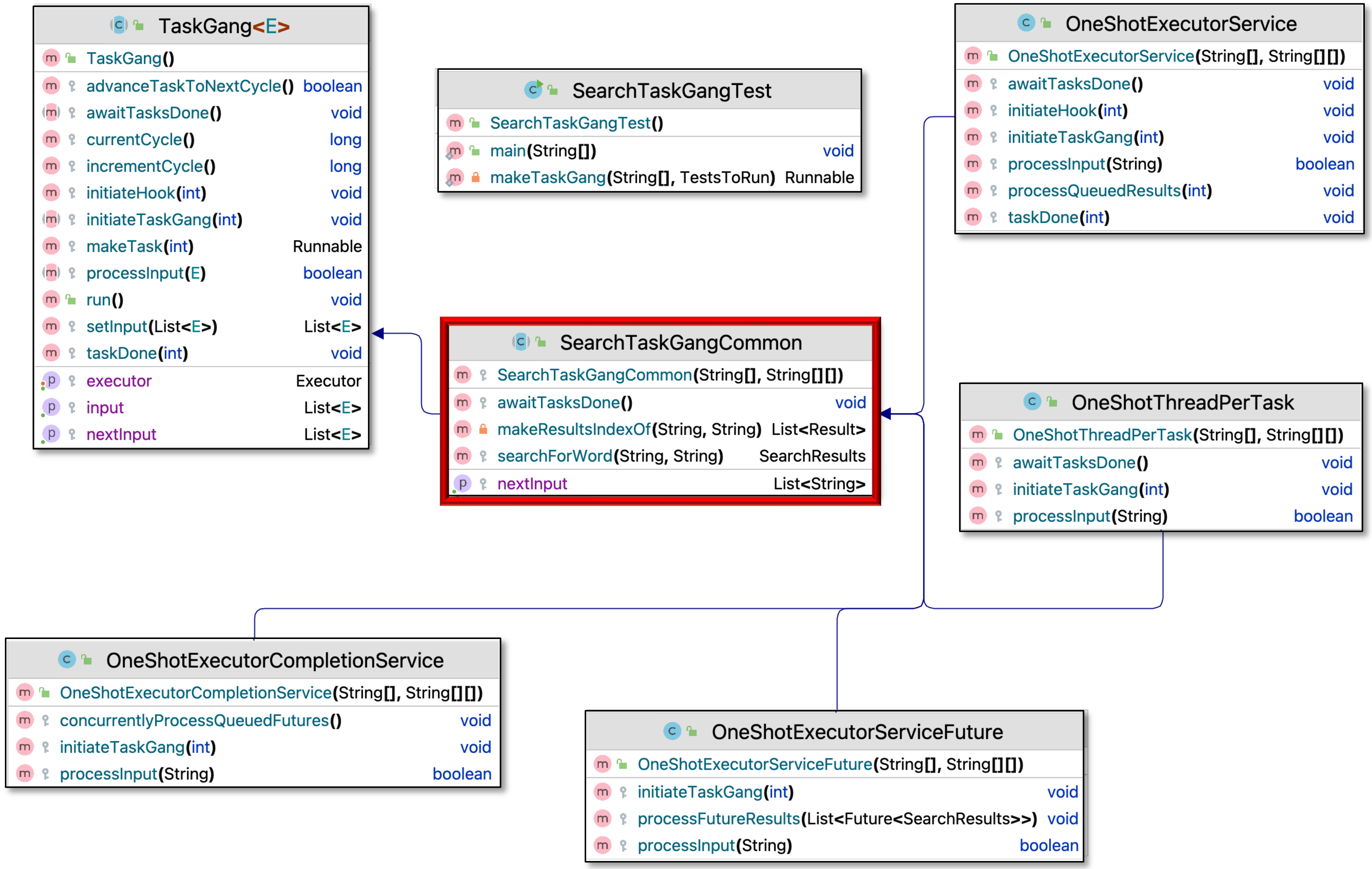
SearchTaskGangCommon

m	SearchTaskGangCommon(String[], String[][])	
f	mInputIterator	Iterator<String[]>
f	mWordsToFind	String[]
m	awaitTasksDone()	void
m	getNextInput()	List<String>
m	makeResultsIndexOf(String, String)	List<Result>
m	searchForWord(String, String)	SearchResults

See SearchTaskGang/src/main/java/tasks/SearchTaskGangCommon.java

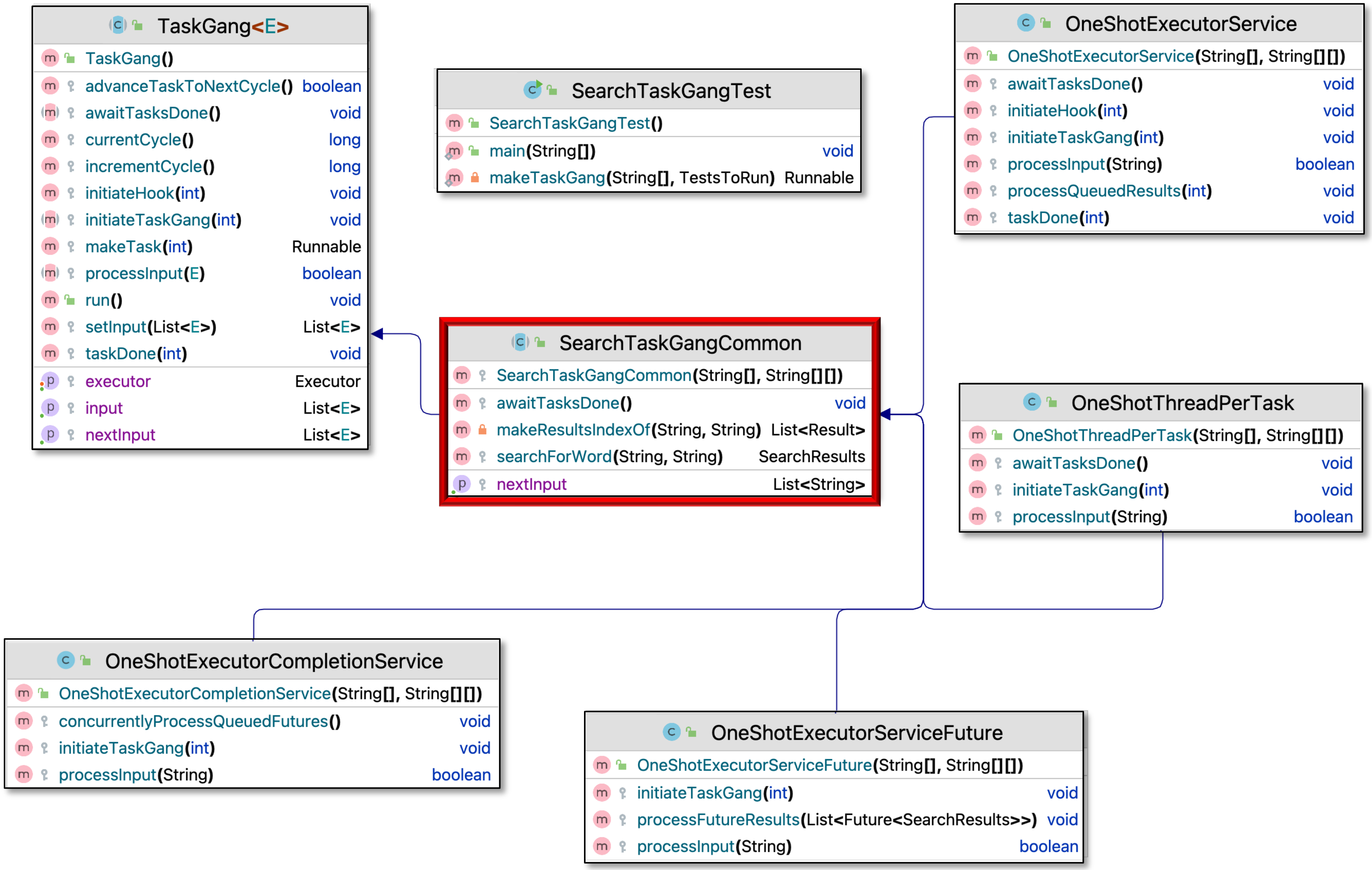
Overview of the Search TaskGangCommon Class

Overview of the SearchTaskGangCommon Class


TaskGang<E>
TaskGang()
advanceTaskToNextCycle() boolean
awaitTasksDone() void
currentCycle() long
incrementCycle() long
initiateHook(int) void
initiateTaskGang(int) void
makeTask(int) Runnable
processInput(E) boolean
run() void
setInput(List<E>) List<E>
taskDone(int) void
executor Executor
input List<E>
nextInput List<E>
SearchTaskGangTest
SearchTaskGangTest()
main(String[]) void
makeTaskGang(String[], TestsToRun) Runnable
OneShotExecutorService
OneShotExecutorService(String[], String[][])
awaitTasksDone() void
initiateHook(int) void
initiateTaskGang(int) void
processInput(String) boolean
processQueuedResults(int) void
taskDone(int) void
SearchTaskGangCommon
SearchTaskGangCommon(String[], String[][])
awaitTasksDone() void
makeResultsIndexOf(String, String) List<Result>
searchForWord(String, String) SearchResults
nextInput List<String>
OneShotThreadPerTask
OneShotThreadPerTask(String[], String[][])
awaitTasksDone() void
initiateTaskGang(int) void
processInput(String) boolean
OneShotExecutorCompletionService
OneShotExecutorCompletionService(String[], String[][])
concurrentlyProcessQueuedFutures() void
initiateTaskGang(int) void
processInput(String) boolean
OneShotExecutorServiceFuture
OneShotExecutorServiceFuture(String[], String[][])
initiateTaskGang(int) void
processFutureResults(List<Future<SearchResults>>) void
processInput(String) boolean

Begins to customize the TaskGang framework so it can concurrently search for keywords in List(s) of String objects

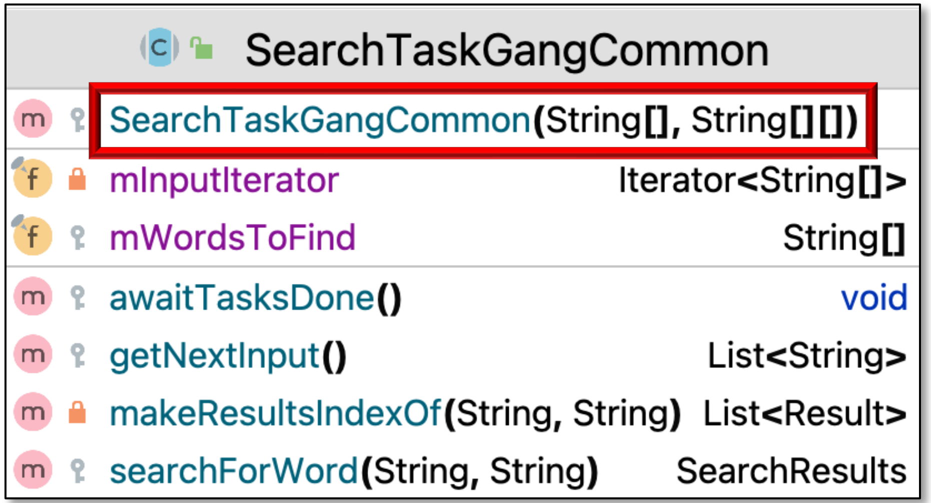
Overview of the SearchTaskGangCommon Class


TaskGang<E>
TaskGang()
advanceTaskToNextCycle() boolean
awaitTasksDone() void
currentCycle() long
incrementCycle() long
initiateHook(int) void
initiateTaskGang(int) void
makeTask(int) Runnable
processInput(E) boolean
run() void
setInput(List<E>) List<E>
taskDone(int) void
executor Executor
input List<E>
nextInput List<E>
SearchTaskGangTest
SearchTaskGangTest()
main(String[]) void
makeTaskGang(String[], TestsToRun) Runnable
OneShotExecutorService
OneShotExecutorService(String[], String[][])
awaitTasksDone() void
initiateHook(int) void
initiateTaskGang(int) void
processInput(String) boolean
processQueuedResults(int) void
taskDone(int) void
SearchTaskGangCommon
SearchTaskGangCommon(String[], String[][])
awaitTasksDone() void
makeResultsIndexOf(String, String) List<Result>
searchForWord(String, String) SearchResults
nextInput List<String>
OneShotThreadPerTask
OneShotThreadPerTask(String[], String[][])
awaitTasksDone() void
initiateTaskGang(int) void
processInput(String) boolean
OneShotExecutorCompletionService
OneShotExecutorCompletionService(String[], String[][])
concurrentlyProcessQueuedFutures() void
initiateTaskGang(int) void
processInput(String) boolean
OneShotExecutorServiceFuture
OneShotExecutorServiceFuture(String[], String[][])
initiateTaskGang(int) void
processFutureResults(List<Future<SearchResults>>) void
processInput(String) boolean

See SearchTaskGang/src/main/java/tasks/SearchTaskGangCommon.java

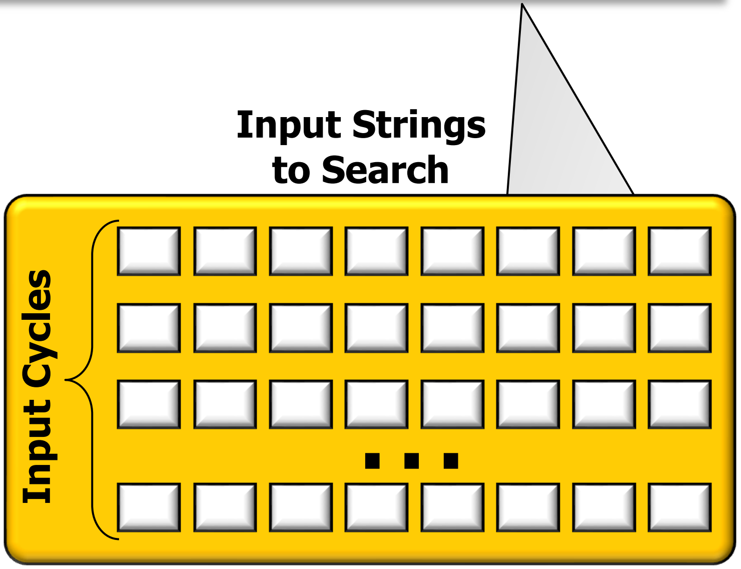
Overview of the SearchTaskGangCommon Class

- TaskGang subclass factors out code common to all the SearchTaskGang Test classes



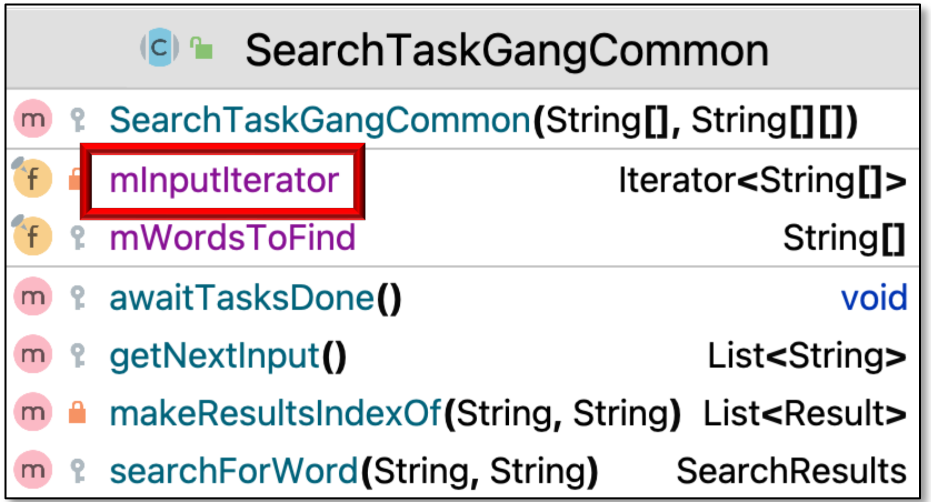
SearchTaskGangCommon

SearchTaskGangCommon(String[], String[][])	
mInputIterator	Iterator<String[]>
mWordsToFind	String[]
awaitTasksDone()	void
getNextInput()	List<String>
makeResultsIndexOf(String, String)	List<Result>
searchForWord(String, String)	SearchResults

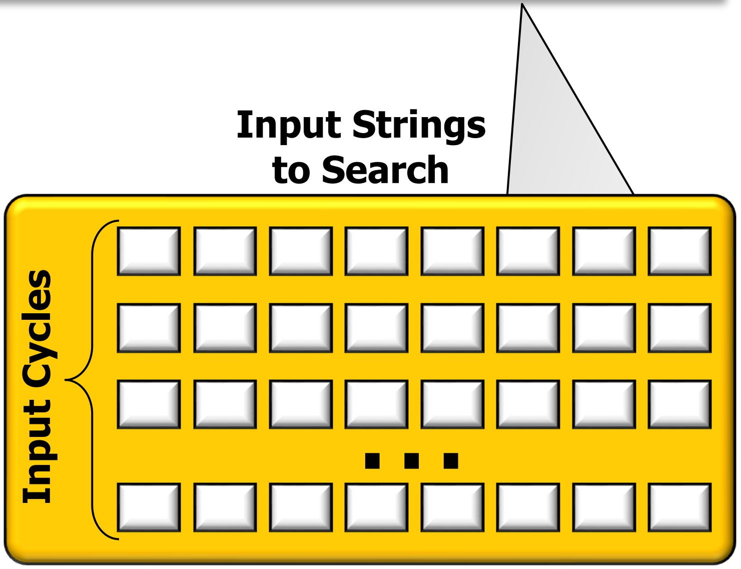


Overview of the SearchTaskGangCommon Class

- TaskGang subclass factors out code common to all the SearchTaskGang Test classes
  - Uses an Iterator to systematically access the input

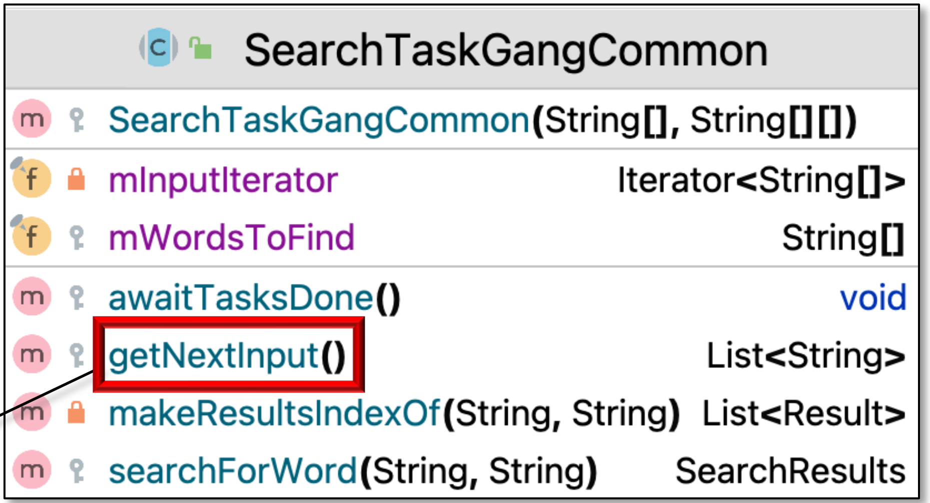


```
SearchTaskGangCommon
m  SearchTaskGangCommon(String[], String[][])
f  mInputIterator                         Iterator<String[]>
f  mWordsToFind                           String[]
m  awaitTasksDone()                       void
m  getNextInput()                         List<String>
m  makeResultsIndexOf(String, String)     List<Result>
m  searchForWord(String, String)          SearchResults
```



Overview of the SearchTaskGangCommon Class

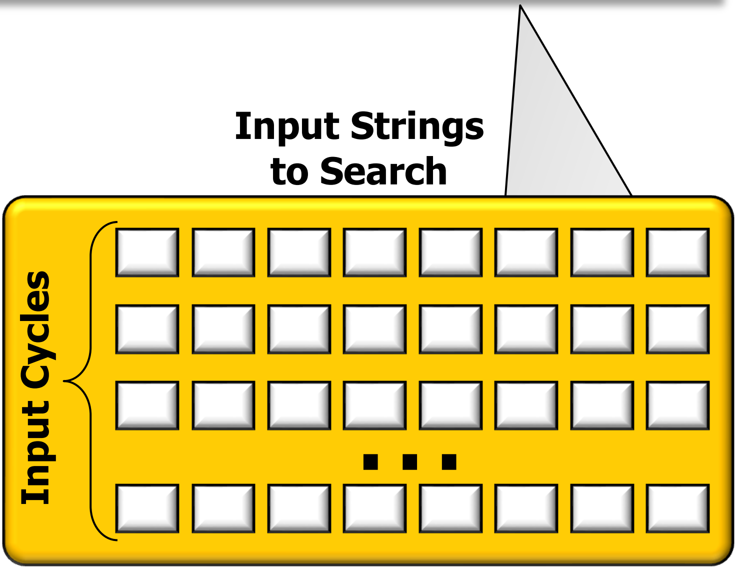
- TaskGang subclass factors out code common to all the SearchTaskGang Test classes
  - Uses an Iterator to systematically access the input
    - Converts array into List



SearchTaskGangCommon

SearchTaskGangCommon(String[], String[][])	
mInputIterator	Iterator<String[]>
mWordsToFind	String[]
awaitTasksDone()	void
getNextInput()	List<String>
makeResultsIndexOf(String, String)	List<Result>
searchForWord(String, String)	SearchResults

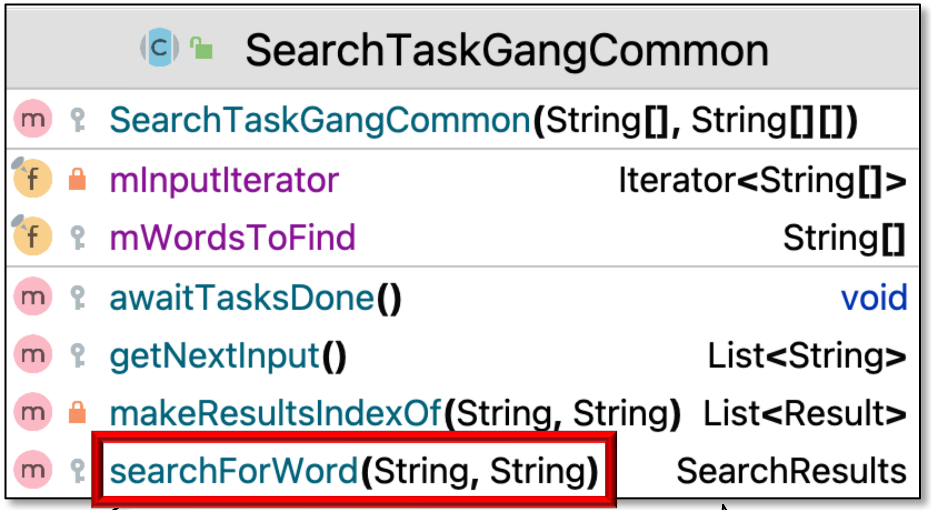
```
if (mInputIterator.hasNext()) {
  mCurrentCycle.
      incrementAndGet();
  return Arrays.asList
      (mInputIterator.next());
} else
  return null;
}
```



The SearchTaskGang case study just uses a single set of input String objects

Overview of the SearchTaskGangCommon Class

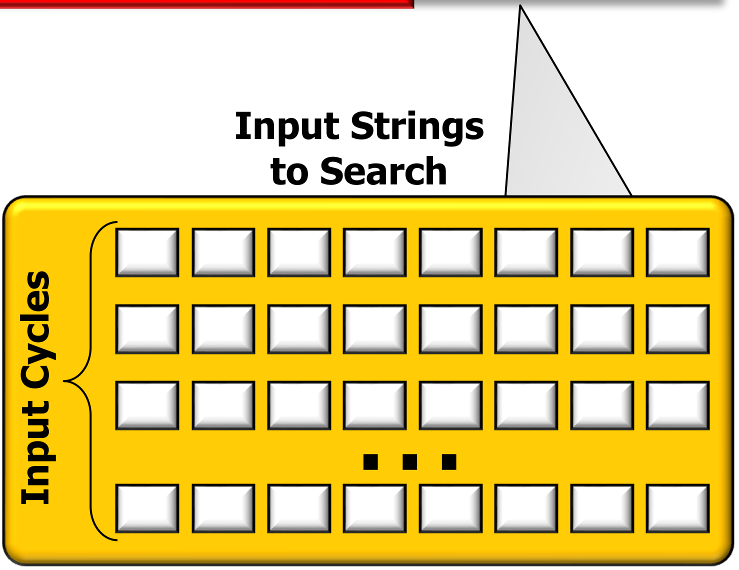
- TaskGang subclass factors out code common to all the SearchTaskGang Test classes
  - Uses an Iterator to systematically access the input
  - Each task runs same logic
    - i.e., returns search results



SearchTaskGangCommon

m	SearchTaskGangCommon(String[], String[][])	
f	mInputIterator	Iterator<String[]>
f	mWordsToFind	String[]
m	awaitTasksDone()	void
m	getNextInput()	List<String>
m	makeResultsIndexOf(String, String)	List<Result>
m	searchForWord(String, String)	SearchResults

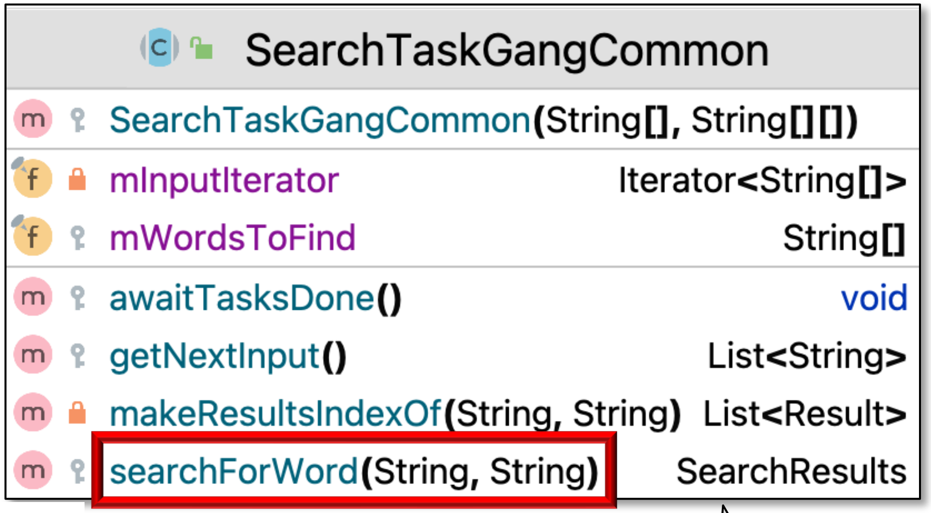
```
// Check to see how many times
// (if any) the word appears
// in the input data.
return new SearchResults(...);
```



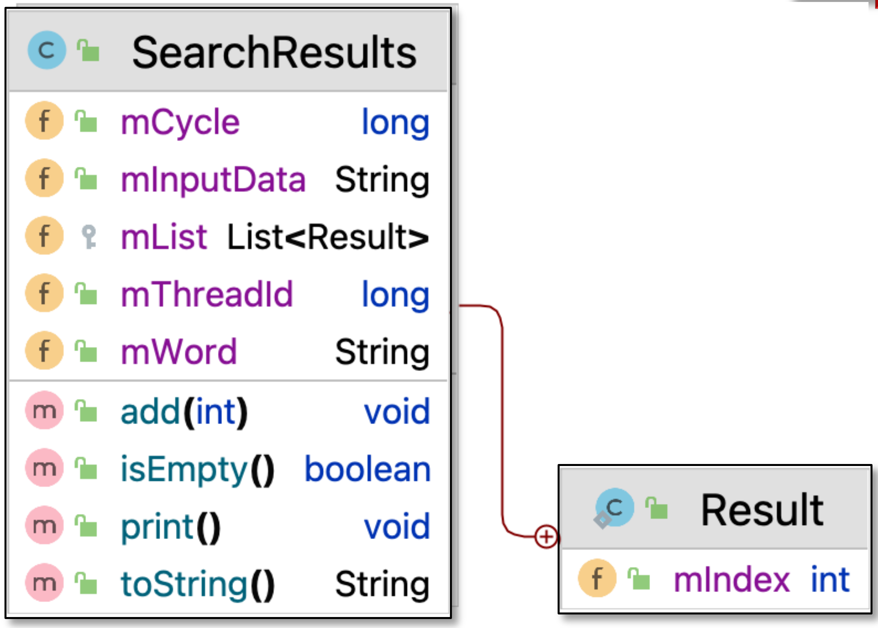
These tasks are "embarrassingly parallel" since there are no dependencies

Overview of the SearchTaskGangCommon Class

- TaskGang subclass factors out code common to all the SearchTaskGang Test classes
  - Uses an Iterator to systematically access the input
  - Each task runs same logic
    - i.e., returns search results


SearchTaskGangCommon

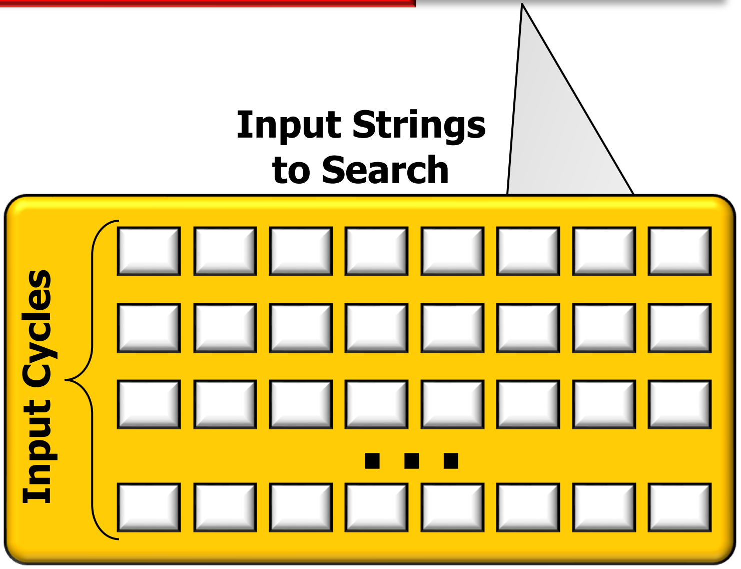
SearchTaskGangCommon(String[], String[][])	
mInputIterator	Iterator<String[]>
mWordsToFind	String[]
awaitTasksDone()	void
getNextInput()	List<String>
makeResultsIndexOf(String, String)	List<Result>
searchForWord(String, String)	SearchResults


SearchResults

mCycle	long
mInputData	String
mList	List<Result>
mThreadId	long
mWord	String
add(int)	void
isEmpty()	boolean
print()	void
toString()	String

Result

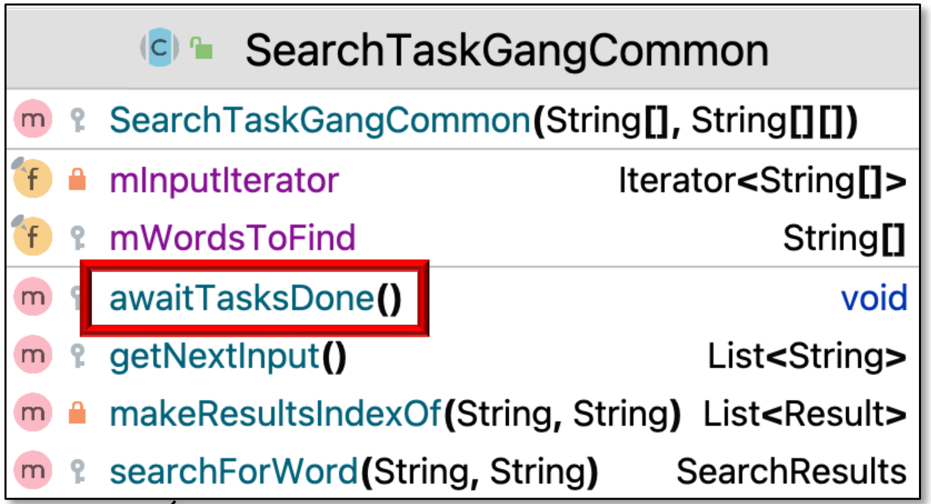
mIndex	int



See SearchTaskGang/src/main/java/utils/SearchResults.java

Overview of the SearchTaskGangCommon Class

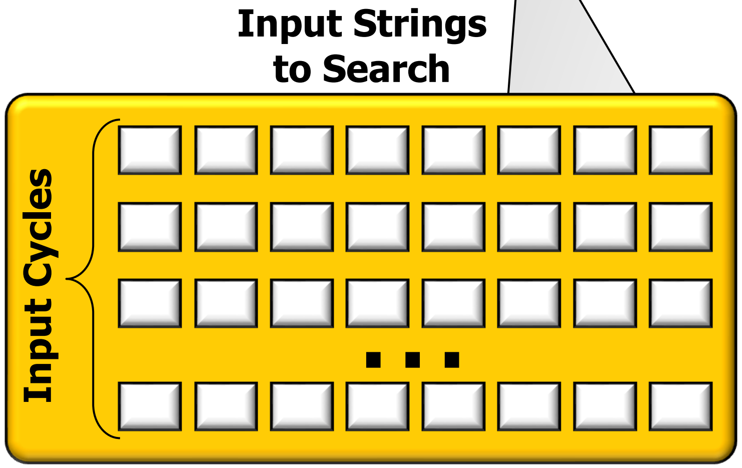
- TaskGang subclass factors out code common to all the SearchTaskGang Test classes
  - Uses an Iterator to systematically access the input
  - Each task runs same logic
  - Barrier shutdowns the Executor & wait for the gang of Threads in the pool to exit



SearchTaskGangCommon

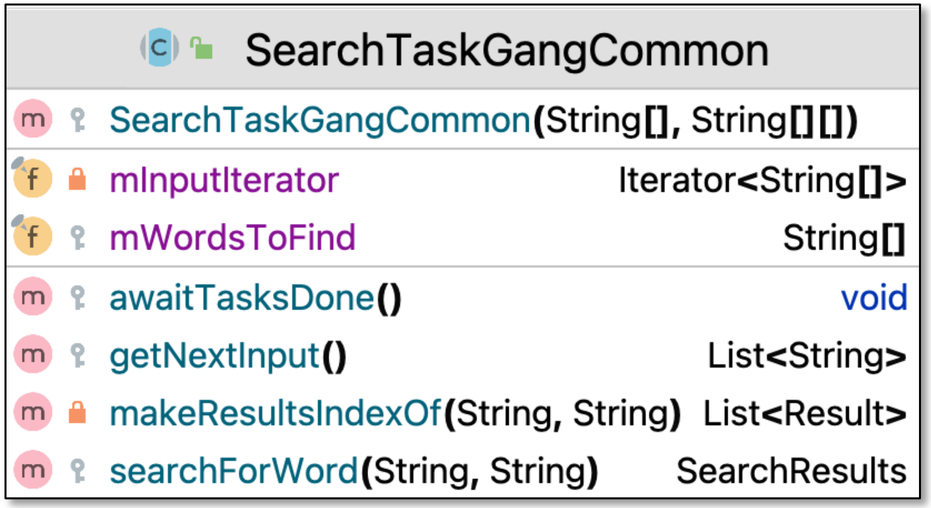
SearchTaskGangCommon(String[], String[][])	
mInputIterator	Iterator<String[]>
mWordsToFind	String[]
awaitTasksDone()	void
getNextInput()	List<String>
makeResultsIndexOf(String, String)	List<Result>
searchForWord(String, String)	SearchResults

```
getExecutor().shutdown();
...
getExecutor().
  awaitTermination(...);
```



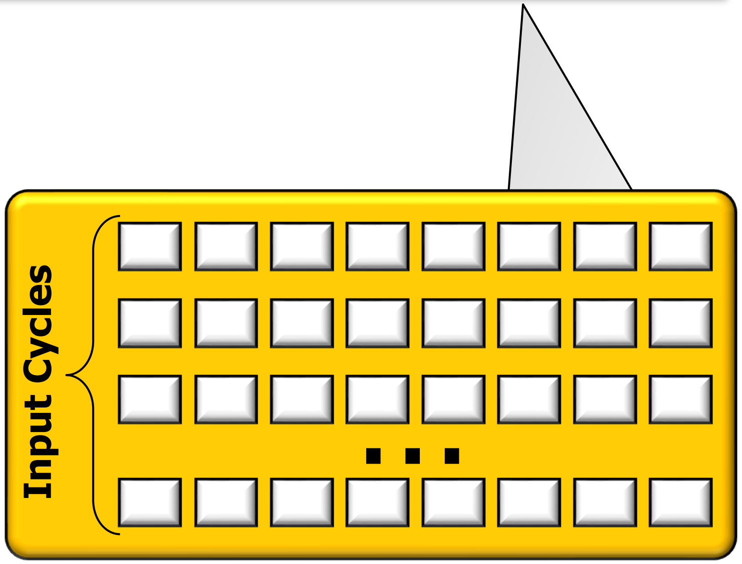
Overview of the SearchTaskGangCommon Class

- There are no commitments (yet) to many of the hook methods defined by the TaskGang framework



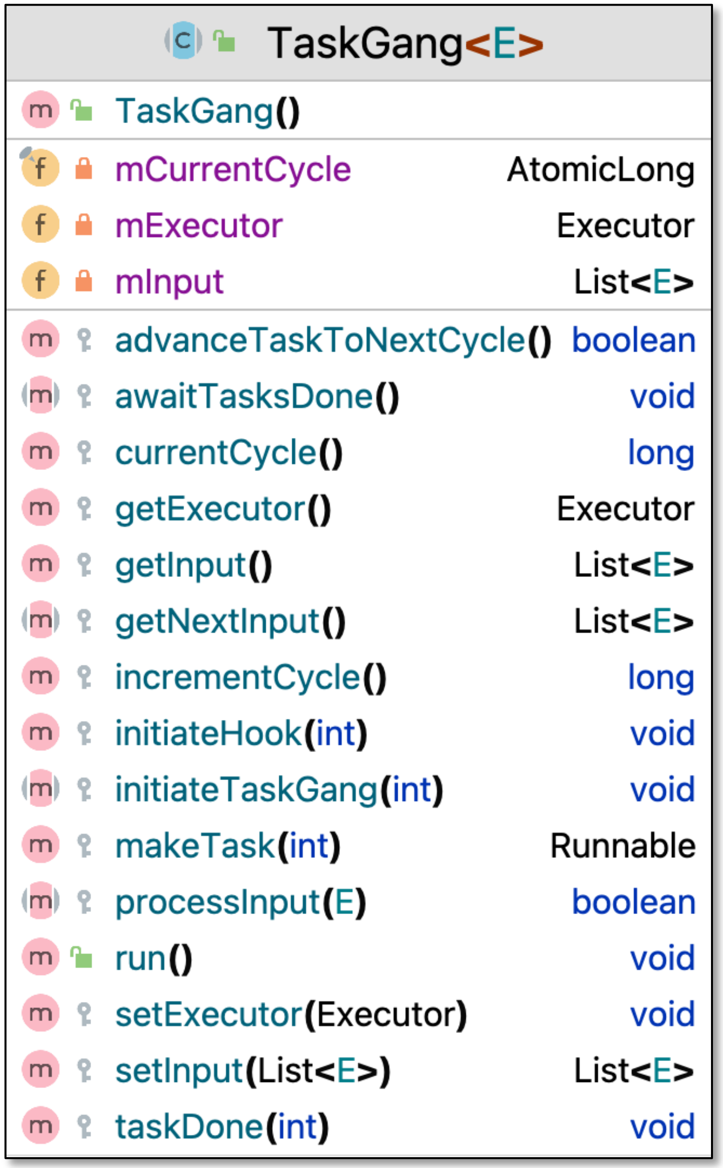
SearchTaskGangCommon

SearchTaskGangCommon(String[], String[][])	
mInputIterator	Iterator<String[]>
mWordsToFind	String[]
awaitTasksDone()	void
getNextInput()	List<String>
makeResultsIndexOf(String, String)	List<Result>
searchForWord(String, String)	SearchResults



Overview of the SearchTaskGangCommon Class

- There are no commitments (yet) to many of the hook methods defined by the TaskGang framework
  - e.g., no Executor implementation, concurrency model, sync vs. async processing, specific source of input Strings, etc.

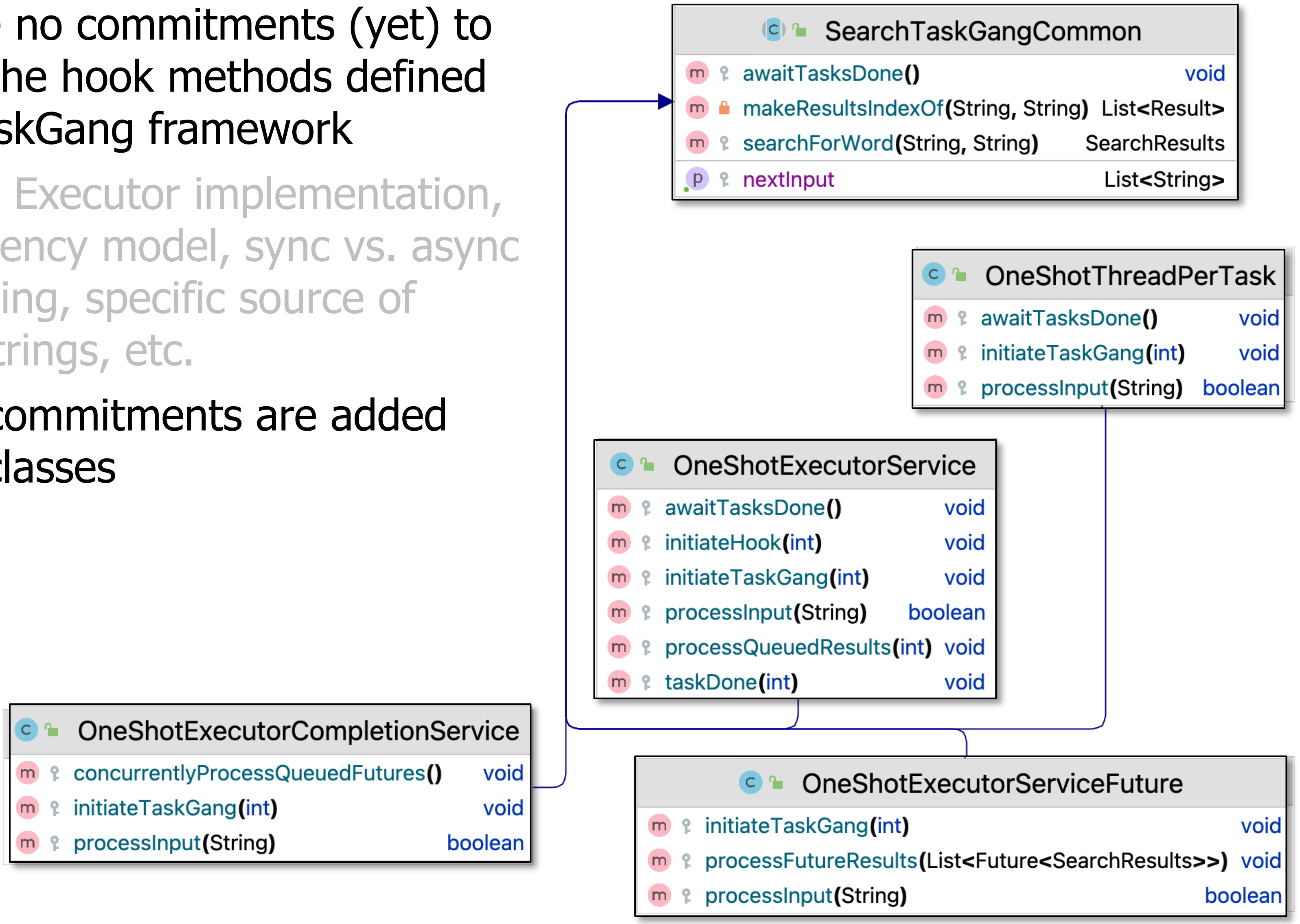


TaskGang<E>

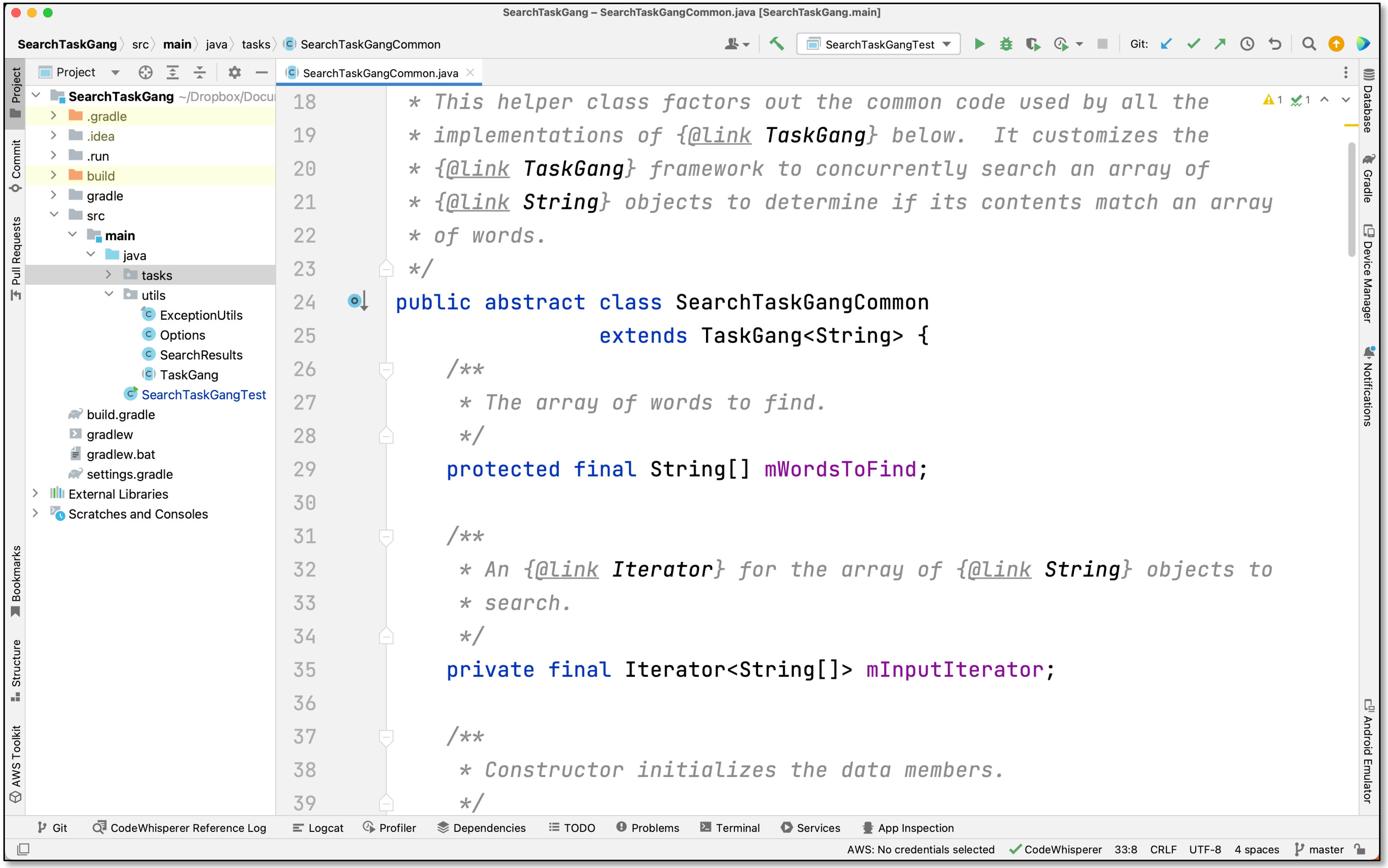
TaskGang()	
mCurrentCycle	AtomicLong
mExecutor	Executor
mInput	List<E>
advanceTaskToNextCycle()	boolean
awaitTasksDone()	void
currentCycle()	long
getExecutor()	Executor
getInput()	List<E>
getNextInput()	List<E>
incrementCycle()	long
initiateHook(int)	void
initiateTaskGang(int)	void
makeTask(int)	Runnable
processInput(E)	boolean
run()	void
setExecutor(Executor)	void
setInput(List<E>)	List<E>
taskDone(int)	void

Overview of the SearchTaskGangCommon Class

- There are no commitments (yet) to many of the hook methods defined by the TaskGang framework
  - e.g., no Executor implementation, concurrency model, sync vs. async processing, specific source of input Strings, etc.
  - These commitments are added by subclasses


SearchTaskGangCommon
awaitTasksDone() void
makeResultsIndexOf(String, String) List<Result>
searchForWord(String, String) SearchResults
nextInput List<String>
OneShotThreadPerTask
awaitTasksDone() void
initiateTaskGang(int) void
processInput(String) boolean
OneShotExecutorService
awaitTasksDone() void
initiateHook(int) void
initiateTaskGang(int) void
processInput(String) boolean
processQueuedResults(int) void
taskDone(int) void
OneShotExecutorCompletionService
concurrentlyProcessQueuedFutures() void
initiateTaskGang(int) void
processInput(String) boolean
OneShotExecutorServiceFuture
initiateTaskGang(int) void
processFutureResults(List<Future<SearchResults>>) void
processInput(String) boolean

Walkthrough of the SearchTaskGangCommon Class



SearchTaskGang – SearchTaskGangCommon.java [SearchTaskGang.main]

SearchTaskGang 〉 src 〉 main 〉 java 〉 tasks 〉 SearchTaskGangCommon

```
SearchTaskGang ~/Dropbox/Docu
  .gradle
  .idea
  .run
  build
  gradle
  src
    main
      java
        tasks
        utils
          ExceptionUtils
          Options
          SearchResults
          TaskGang
        SearchTaskGangTest
  build.gradle
  gradlew
  gradlew.bat
  settings.gradle
External Libraries
Scratches and Consoles
```

```java
 * This helper class factors out the common code used by all the
 * implementations of {@link TaskGang} below.  It customizes the
 * {@link TaskGang} framework to concurrently search an array of
 * {@link String} objects to determine if its contents match an array
 * of words.
 */
public abstract class SearchTaskGangCommon
                      extends TaskGang<String> {
    /**
     * The array of words to find.
     */
    protected final String[] mWordsToFind;

    /**
     * An {@link Iterator} for the array of {@link String} objects to
     * search.
     */
    private final Iterator<String[]> mInputIterator;

    /**
     * Constructor initializes the data members.
     */
```

See SearchTaskGang/src/main/java/tasks/SearchTaskGangCommon.java

End of Overview of the Search TaskGangCommon Class