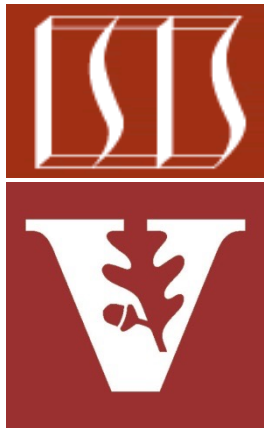


Overview of the OneShot ThreadPerTask Class



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

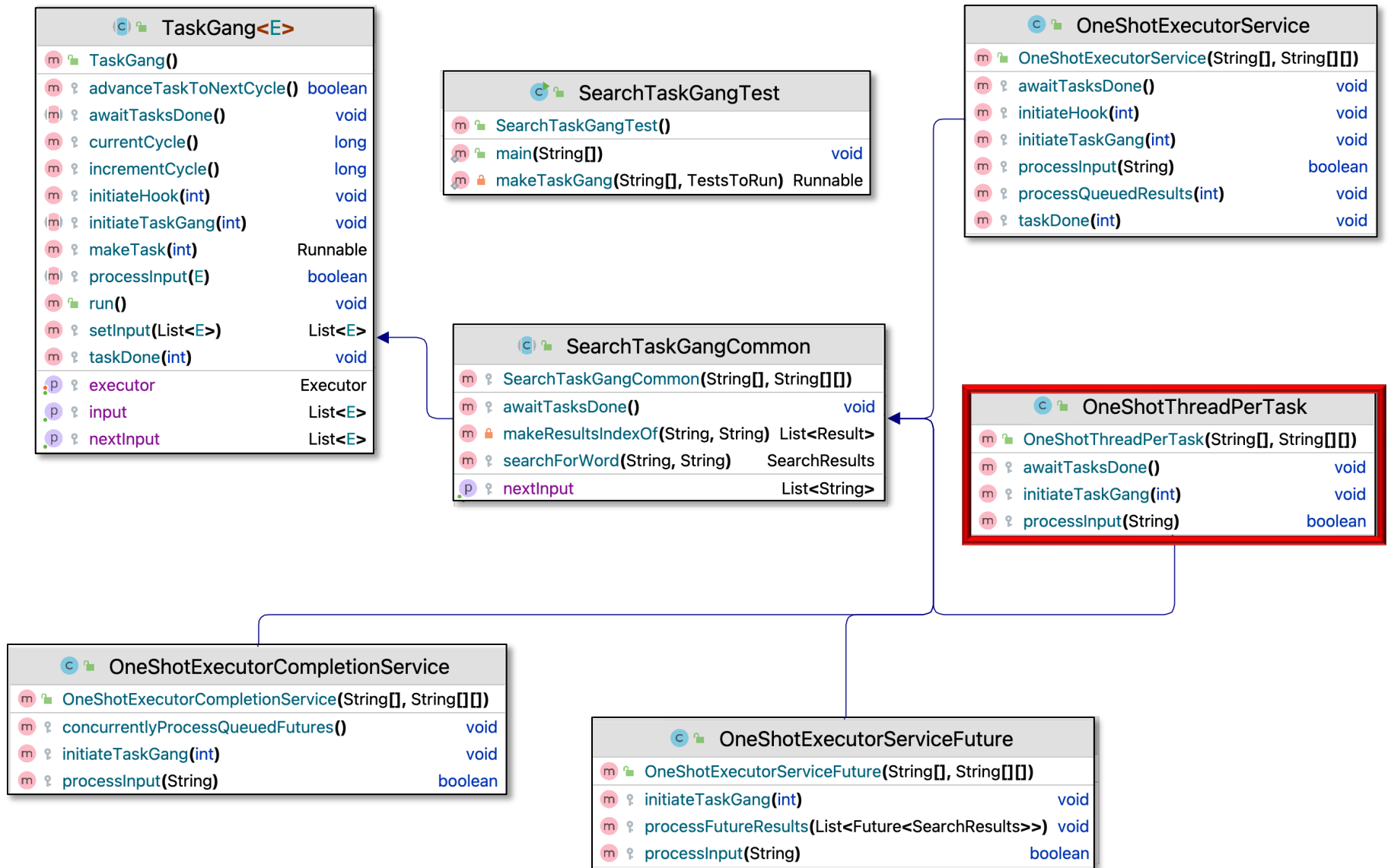
- Understand the SearchTaskGang case study
- Recognize the methods that are defined by the TaskGang framework
- Know the subclasses that extends TaskGang (directly or indirectly)
 - SearchTaskGangCommon
 - OneShotThreadPerTask

OneShotThreadPerTask	
m	OneShotThreadPerTask(String[], String[][])
f	mWorkerThreads List<Thread>
m	awaitTasksDone() void
m	initiateTaskGang(int) void
m	processInput(String) boolean

See [SearchTaskGang/src/main/java/tasks/OneShotThreadPerTask.java](#)

Overview of One ShotThreadPerTask

Overview of OneShotThreadPerTask



Customizes SearchTaskGangCommon to do a one-shot search for words in a List of String objects via virtual Thread objects instead of using a Thread Pool

Overview of OneShotThreadPerTask

TaskGang<E>	
m	TaskGang()
m	advanceTaskToNextCycle() boolean
m	awaitTasksDone() void
m	currentCycle() long
m	incrementCycle() long
m	initiateHook(int) void
m	initiateTaskGang(int) void
m	makeTask(int) Runnable
m	processInput(E) boolean
m	run() void
m	setInput(List<E>) List<E>
m	taskDone(int) void
p	executor Executor
p	input List<E>
p	nextInput List<E>

SearchTaskGangTest	
m	SearchTaskGangTest()
m	main(String[]) void
m	makeTaskGang(String[], TestsToRun) Runnable

SearchTaskGangCommon	
m	SearchTaskGangCommon(String[], String[][]) void
m	awaitTasksDone() void
m	makeResultsIndexof(String, String) List<Result>
m	searchForWord(String, String) SearchResults
p	nextInput List<String>

OneShotExecutorService	
m	OneShotExecutorService(String[], String[][]) void
m	awaitTasksDone() void
m	initiateHook(int) void
m	initiateTaskGang(int) void
m	processInput(String) boolean
m	processQueuedResults(int) void
m	taskDone(int) void

OneShotThreadPerTask	
m	OneShotThreadPerTask(String[], String[][]) void
m	awaitTasksDone() void
m	initiateTaskGang(int) void
m	processInput(String) boolean

OneShotExecutorCompletionService	
m	OneShotExecutorCompletionService(String[], String[][]) void
m	concurrentlyProcessQueuedFutures() void
m	initiateTaskGang(int) void
m	processInput(String) boolean

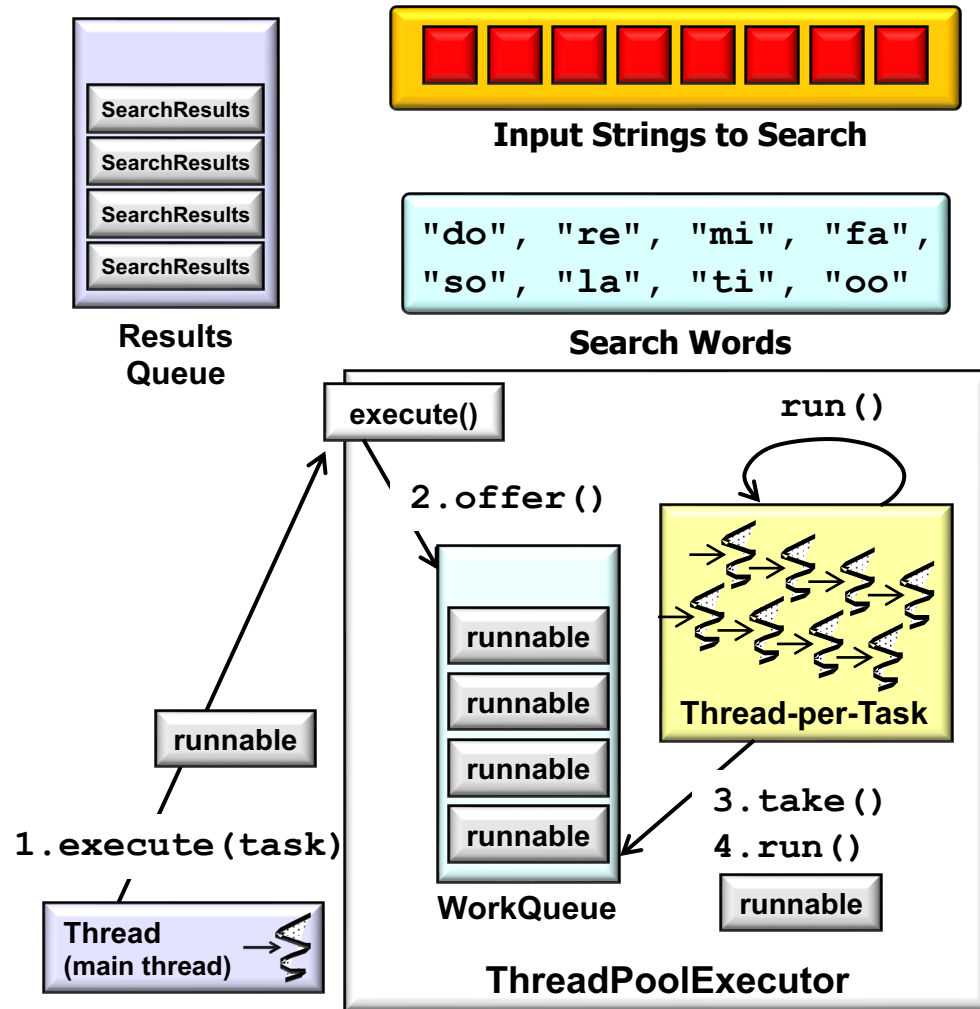
OneShotExecutorServiceFuture	
m	OneShotExecutorServiceFuture(String[], String[][]) void
m	initiateTaskGang(int) void
m	processFutureResults(List<Future<SearchResults>>) void
m	processInput(String) boolean

See [SearchTaskGang/src/main/java/tasks/OneShotThreadPerTask.java](#)

Overview of OneShotThreadPerTask

- Customizes SearchTaskGang
Common to search for words in a List of String objects

Strategy	Implementation
<i>Executor model</i>	Virtual Thread-per-Task
<i>Unit of concurrency</i>	Virtual Thread per input String
<i>Results processing model</i>	Synchronous processing



Overview of OneShotThreadPerTask

- Customizes SearchTaskGang
Common to search for words
in a List of String objects

OneShotThreadPerTask		
m	OneShotThreadPerTask(String[], String[][])	
f	mWorkerThreads	List<Thread>
m	awaitTasksDone()	void
m	initiateTaskGang(int)	void
m	processInput(String)	boolean

Overview of OneShotThreadPerTask

- Customizes SearchTaskGang
Common to search for words
in a List of String objects
- Creates an Executor to start
a virtual Thread-per-task

OneShotThreadPerTask		
m	OneShotThreadPerTask(String[], String[][])	
f	mWorkerThreads	List<Thread>
m	awaitTasksDone()	void
m	initiateTaskGang(int)	void
m	processInput(String)	boolean

```
if (getExecutor() == null)
    setExecutor (runnable -> mWorkerThreads
                .add(Thread.startVirtualThread(runnable))) ;

for (int i = 0; i < inputSize; ++i)
    getExecutor().execute(makeTask(i));
```


Overview of OneShotThreadPerTask

- Customizes SearchTaskGang
Common to search for words
in a List of String objects
- Creates an Executor to start
a virtual Thread-per-task
 - Each virtual Thread is
stored in a List

OneShotThreadPerTask		
m	OneShotThreadPerTask(String[], String[][])	
f	mWorkerThreads	List<Thread>
m	awaitTasksDone()	void
m	initiateTaskGang(int)	void
m	processInput(String)	boolean

```
if (getExecutor() == null)
    setExecutor (runnable -> mWorkerThreads
                .add(Thread.startVirtualThread(runnable)) );

for (int i = 0; i < inputSize; ++i)
    getExecutor().execute(makeTask(i));
```

Overview of OneShotThreadPerTask

- Customizes SearchTaskGang
Common to search for words in a List of String objects
- Creates an Executor to start a virtual Thread-per-task
- Word searches run in a dedicated virtual Thread

OneShotThreadPerTask		
m	OneShotThreadPerTask(String[], String[][])	
f	mWorkerThreads	List<Thread>
m	awaitTasksDone()	void
m	initiateTaskGang(int)	void
m	processInput(String)	boolean

```
for (String word : mWordsToFind) {  
    var results = searchForWord(word, inputData);  
  
    synchronized(System.out) {  
        results.print();  
    }  
}
```

Overview of OneShotThreadPerTask

- Customizes SearchTaskGang
Common to search for words
in a List of String objects
 - Creates an Executor to start
a virtual Thread-per-task
 - Word searches run in a
dedicated virtual Thread
 - The main Thread joins all the
worker virtual Thread objects

OneShotThreadPerTask		
m	OneShotThreadPerTask(String[], String[][])	
f	mWorkerThreads	List<Thread>
m	awaitTasksDone()	void
m	initiateTaskGang(int)	void
m	processInput(String)	boolean

```
for (Thread thread : mWorkerThreads)
    rethrowRunnable(thread::join);
```

Walkthrough the OneShotThreadPerTask

```
SearchTaskGang - OneShotThreadPerTask.java [SearchTaskGang.main]
SearchTaskGang src main java tasks
Project SearchTaskGang ~/Dropbox/Doc
  .gradle
  .idea
  .run
  build
  gradle
  src
    main
      java
        tasks
          utils
            SearchTaskGangTest
        build.gradle
        gradlew
        gradlew.bat
        settings.gradle
  External Libraries
  Scratches and Consoles
12 /**
13  * Customizes the {@link SearchTaskGangCommon} framework to process a
14  * one-shot {@link List} of tasks via an {@link Executor} that creates
15  * a virtual {@link Thread} for each task. The {@link Executor} model
16  * uses a virtual {@link Thread}-per-task. The unit of concurrency is
17  * each input {@link String} and the results processing model is
18  * synchronous.
19  */
20 public class OneShotThreadPerTask
21     extends SearchTaskGangCommon {
22     /**
23      * The {@link List} of worker {@link Thread} objects that were
24      * created.
25      */
26     private final List<Thread> mWorkerThreads;
27
28     /**
29      * Constructor initializes the superclass and data members.
30      */
31     public OneShotThreadPerTask(String[] wordsToFind,
32                                 String[][] stringsToSearch) {
33         // Pass input to superclass constructor.
```

See [SearchTaskGang/src/main/java/tasks/OneShotThreadPerTask.java](#)

End of Overview of OneShotThreadPerTask