

Overview of the Java Executor Framework

(Part 2)

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

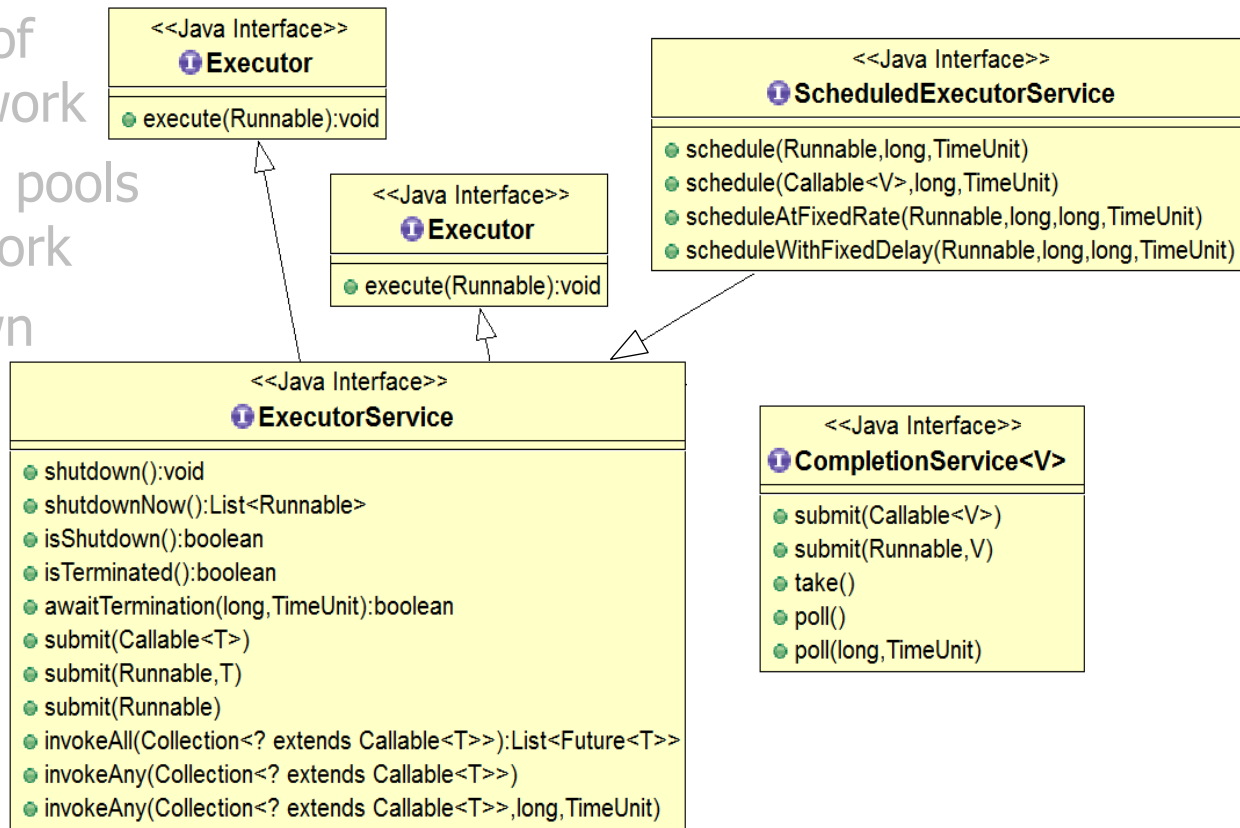
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

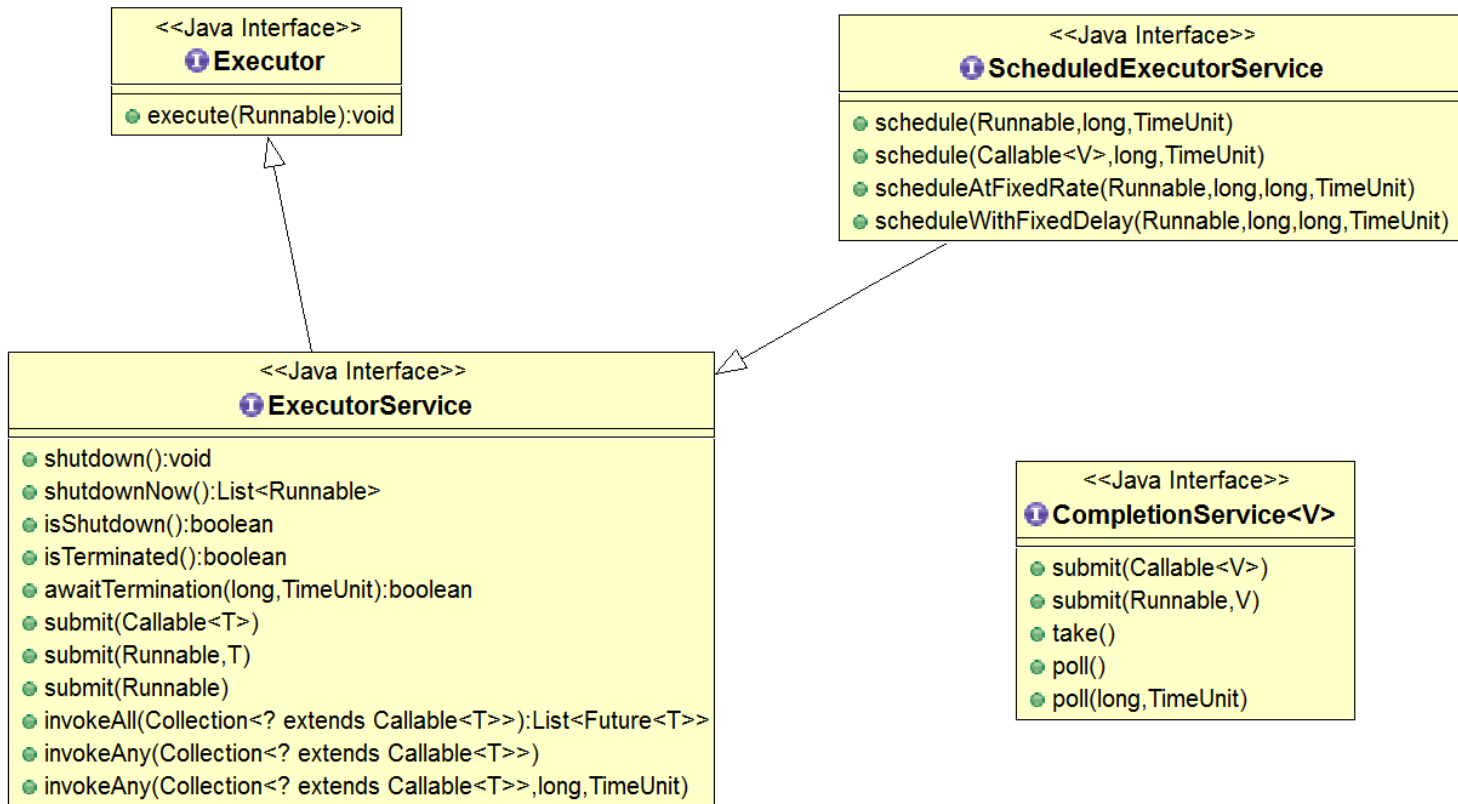
- Understand the purpose of the Java executor framework
- Know the types of thread pools supported by the framework
- Recognize a human known use of thread pools
- Learn the key interfaces the framework provides



Key Java Executor Framework Interfaces

Key Java Executor Framework Interfaces

- The Java executor framework contains several key interfaces



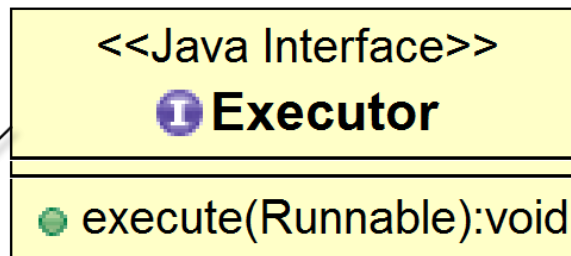
See docs.oracle.com/javase/tutorial/essential/concurrency/executors.html

Key Java Executor Framework Interfaces

- The Java executor framework contains several key interfaces

- **Executor**

- Provides a means of submitting new runnable tasks for execution



Defines a simple API that decouples task submission from the mechanics of how each task will be run

Key Java Executor Framework Interfaces

- The Java executor framework contains several key interfaces

- **Executor**

- **ExecutorService**

- Extends Executor to manage task & executor lifecycles



<<Java Interface>>

ExecutorService

- shutdown():void
- shutdownNow():List<Runnable>
- isShutdown():boolean
- isTerminated():boolean
- awaitTermination(long,TimeUnit):boolean
- submit(Callable<T>):Future<T>
- submit(Runnable,T):Future<T>
- submit(Runnable):Future<?>
- invokeAll(Collection<? extends Callable<T>>):List<Future<T>>
- invokeAny(Collection<? extends Callable<T>>)
- invokeAny(Collection<? extends Callable<T>>,long,TimeUnit)

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html

Key Java Executor Framework Interfaces

- The Java executor framework contains several key interfaces

- **Executor**
- **ExecutorService**
- **ScheduledExecutorService**
 - Extends ExecutorService to support future and/or periodic execution of tasks

<<Java Interface>>

ScheduledExecutorService

- `schedule(Runnable,long,TimeUnit)`
- `schedule(Callable<V>,long,TimeUnit)`
- `scheduleAtFixedRate(Runnable,long,long,TimeUnit)`
- `scheduleWithFixedDelay(Runnable,long,long,TimeUnit)`



See docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html

Key Java Executor Framework Interfaces

- The Java executor framework contains several key interfaces

- Executor**
- ExecutorService**
- ScheduledExecutorService**

- CompletionService**

- Decouples task invocation from processing of completed task results



<<Java Interface>>

CompletionService<V>

- submit(Callable<V>):Future<V>
- submit(Runnable,V):Future<V>
- take():Future<V>
- poll():Future<V>
- poll(long,TimeUnit):Future<V>

End of Overview of the Java Executor Framework (Part 2)