Overview of the Java Executor Framework (Part 2)

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 how the Java executor framework decouples the creation & management of threads from the rest of the app logic
- Know the types of thread pools supported by the Java executor framework
- Recognize a human known use of thread pools
- Learn the key interfaces provided by the Java executor framework
Key Java Executor Framework Interfaces
The Java executor framework contains several key interfaces:

- **Executor**
  - `execute(Runnable):void`

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

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

- **CompletionService<V>**
  - `submit(Callable<V>)`
  - `submit(Runnable, V)`
  - `take()`
  - `poll()`
  - `poll(long, TimeUnit)`

See [docs.oracle.com/javase/tutorial/essential/concurrency/executors.html](http://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
      - `Executor` interface
        - `execute(Runnable):void`

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html)
Key Java Executor Framework Interfaces

The Java executor framework contains several key interfaces:

- **Executor**
- **ExecutorService**
  - Extends Executor to manage task & executor lifecycles

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

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/ScheduledExecutorService.html](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 asynchronous task invocation from processing of completed task results

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletionService.html
End of Overview of the Java Executor Framework (Part 2)
Overview of the Java Executor Framework (Part 3)

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 how the Java executor framework decouples the creation & management of threads from the rest of the app logic
- Know the types of thread pools supported by the Java executor framework
- Recognize a human known use of thread pools
- Learn the key interfaces provided by the Java executor framework
- Be aware of the factory methods provided by the Java Executors class
The Java Executors Class
The Java Executors Class

- Executors is a utility class that creates executor implementations.

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html
The Java Executors Class

- Executors is a utility class that creates executor implementations
- A utility class is a final class in Java with a private constructor that defines a set of static methods that perform common, reusable functions

See [en.wikipedia.org/wiki/Utility_class](en.wikipedia.org/wiki/Utility_class)
The Java Executors Class

• Executors is a utility class that creates executor implementations
  • A utility class is a final class in Java with a private constructor that defines a set of static methods that perform common, reusable functions
  • Factory methods create desired executors
    • e.g., cached, fixed, work-stealing thread pools, etc.
The Java Executors Class

- Executors is a utility class that creates executor implementations
  - A utility class is a final class in Java with a private constructor that defines a set of static methods that perform common, reusable functions
  - Factory methods create desired executors
  - ThreadFactory creates new threads

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/ThreadFactory.html
The Java Executors Class

- Executors is a utility class that creates executor implementations
- A utility class is a final class in Java with a private constructor that defines a set of static methods that perform common, reusable functions
- Factory methods create desired executors
- ThreadFactory creates new threads
- A default thread factory is provided

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#defaultThreadFactory](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/Executors.html#defaultThreadFactory)
End of Overview of the Java Executors Framework (Part 3)