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
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

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

```
<<Java Interface>>
\begin{itemize}
  \item Executor
\end{itemize}
\begin{itemize}
  \item execute(Runnable):void
\end{itemize}

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

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 consumption of completed task results

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletionService.html
Key Java Executor Framework Interfaces

- The Java executor framework contains several key interfaces
  - `Executor`
  - `ExecutorService`
  - `ScheduledExecutorService`
  - `CompletionService`
    - Decouples async task invocation from completed task results
    - Typically implemented via an executor & a blocking queue
Key Java Executor Framework Interfaces

- The Java executor framework contains several key interfaces
  - Executor
  - ExecutorService
  - ScheduledExecutorService
  - CompletionService

  - Decouples async task invocation from completed task results
  - Implements the Proactor pattern
  - Supports the demultiplexing & dispatching of multiple event handlers, which are triggered by the completion of asynchronous events

See en.wikipedia.org/wiki/Proactor_pattern
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 Java utility class is a final class with a private constructor that defines a set of static methods that perform common, reusable functions

See en.wikipedia.org/wiki/Utility_class
The Java Executors Class

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

![Java Class Diagram]

- `newFixedThreadPool(int): ExecutorService`
- `newWorkStealingPool(int): ExecutorService`
- `newFixedThreadPool(): ExecutorService`
- `newWorkStealingPool(): ExecutorService`
- `newFixedThreadPool(int, ThreadFactory): ExecutorService`
- `newSingleThreadExecutor(): ExecutorService`
- `newSingleThreadExecutor(ThreadFactory): ExecutorService`
- `newCachedThreadPool(): ExecutorService`
- `newCachedThreadPool(ThreadFactory): ExecutorService`
- `newSingleThreadScheduledExecutor(): ScheduledExecutorService`
- `newSingleThreadScheduledExecutor(ThreadFactory): ScheduledExecutorService`
- `newScheduledThreadPool(int): ScheduledExecutorService`
- `newScheduledThreadPool(int, ThreadFactory): ScheduledExecutorService`
- `defaultThreadFactory()`
- `privilegedThreadFactory()`
- `callable(Runnable, T): Callable<T>`
- `callable(Runnable): Callable<Object>`
- `callable(PrivilegedAction<?>): Callable<Object>`
- `callable(PrivilegedExceptionAction<?>): Callable<Object>`
- `privilegedCallable(Callable<T>): Callable<T>`
- `privilegedCallableUsingCurrentClassLoader(Callable<T>): Callable<T>`
The Java Executors Class

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

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

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