Overview of the Java Executor Framework (Part 3)

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

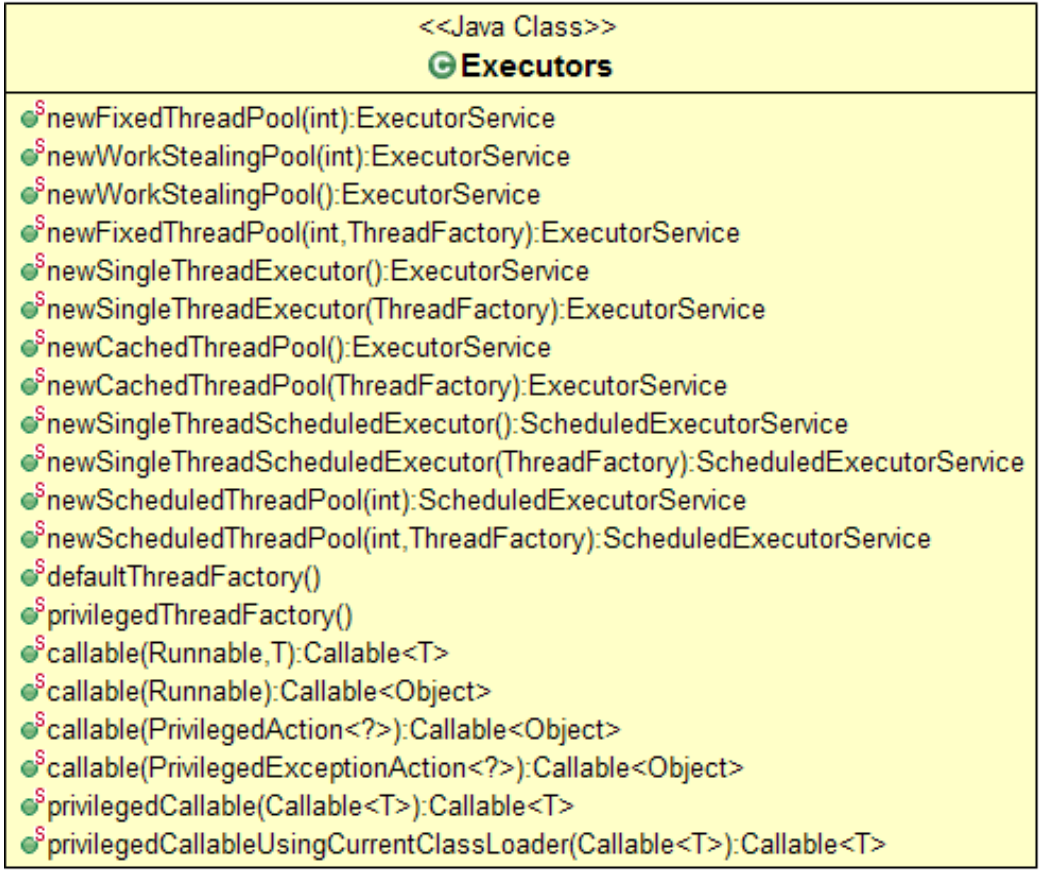
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

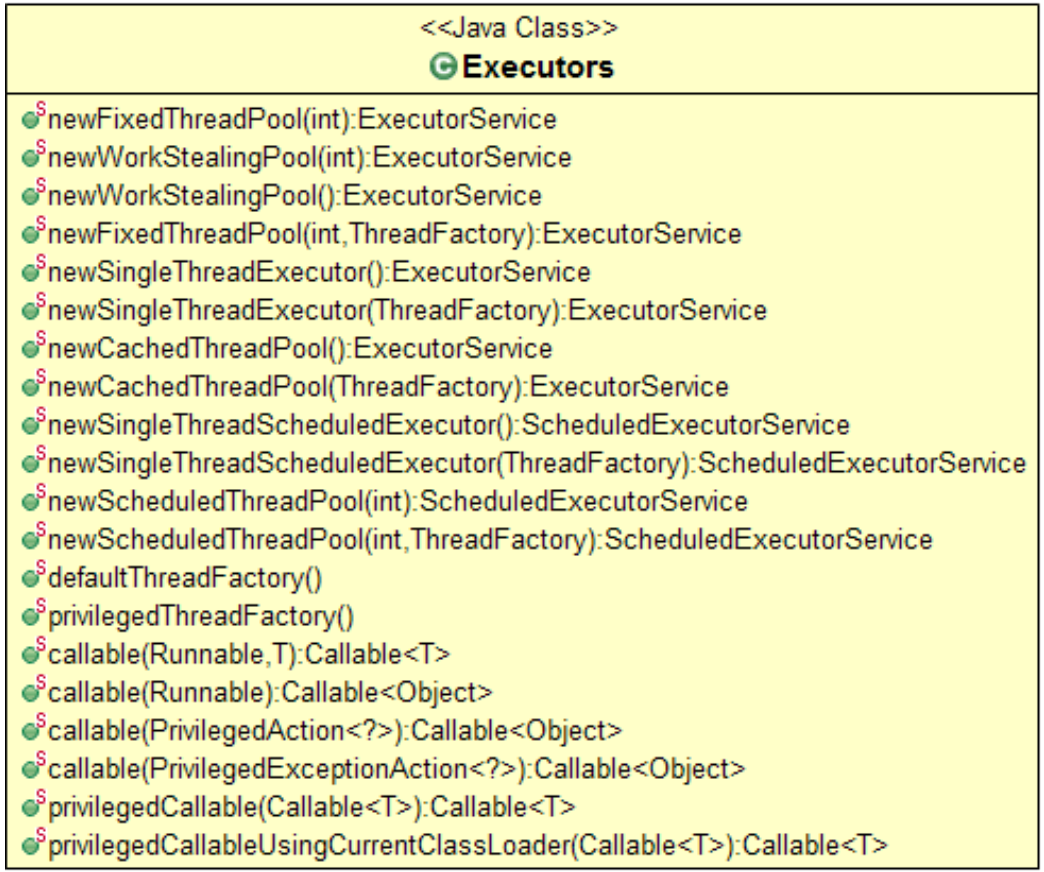
- Understand the purpose the Java executor framework
- Know the types of thread pools supported by the framework
- Recognize a human known use of thread pools
- Learn the key interfaces the framework provides
- Be aware of the factory methods provided by the Java Executors class



The Java Executors Class

The Java Executors Class

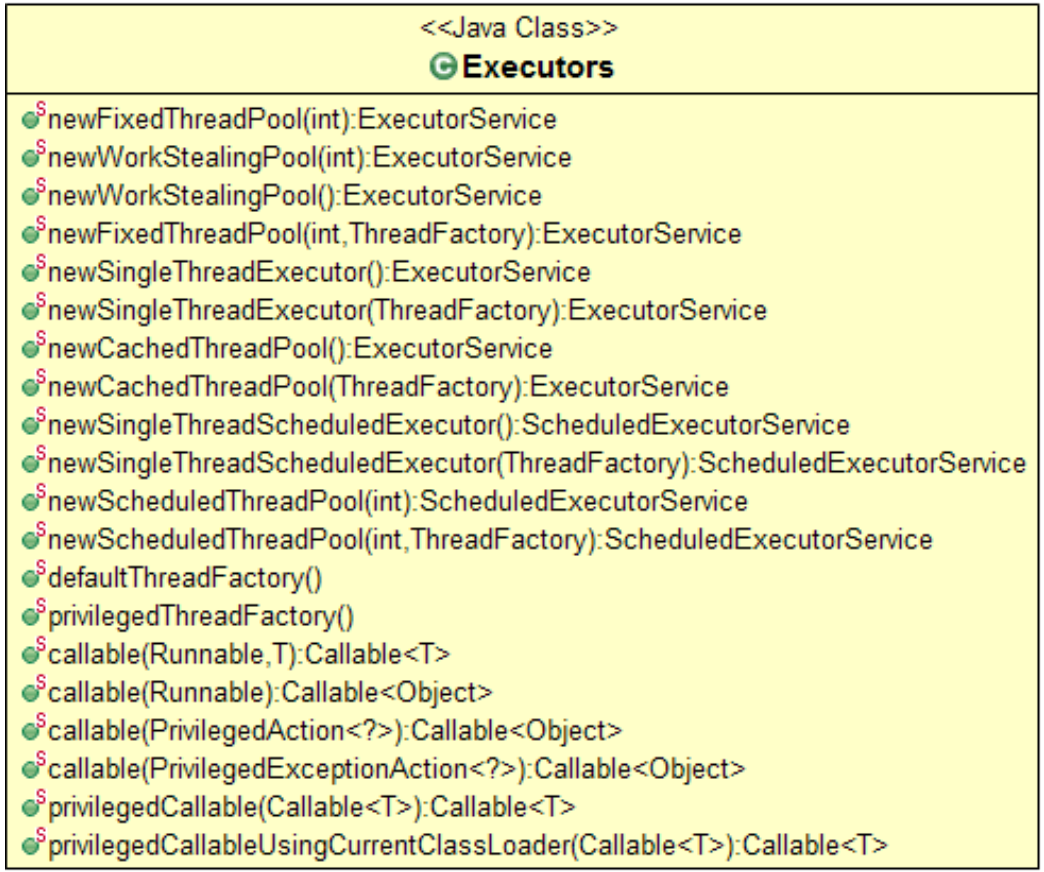
- Executors is a utility class that creates executor implementations



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

The Java Executors Class

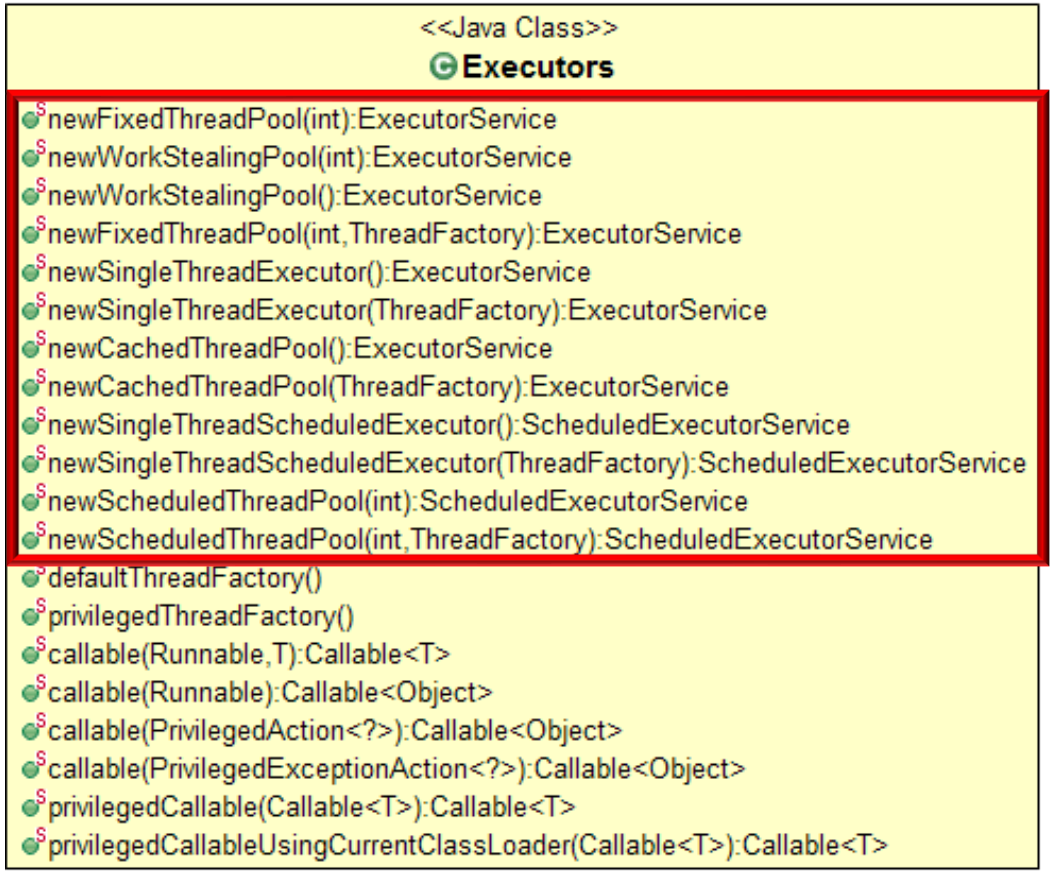
- Executors is a utility class that creates executor implementations
  - A utility class is a final class having only static methods, no state, & a private constructor



See www.quora.com/What-is-the-best-way-to-write-utility-classes-in-Java/answer/Jon-Harley

The Java Executors Class

- Executors is a utility class that creates executor implementations
  - A utility class is a final class having only static methods, no state, & a private constructor
  - Factory methods create desired executors



See en.wikipedia.org/wiki/Factory_method_pattern

The Java Executors Class

- Executors is a utility class that creates executor implementations
  - A utility class is a final class having only static methods, no state, & a private constructor
  - Factory methods create desired executors

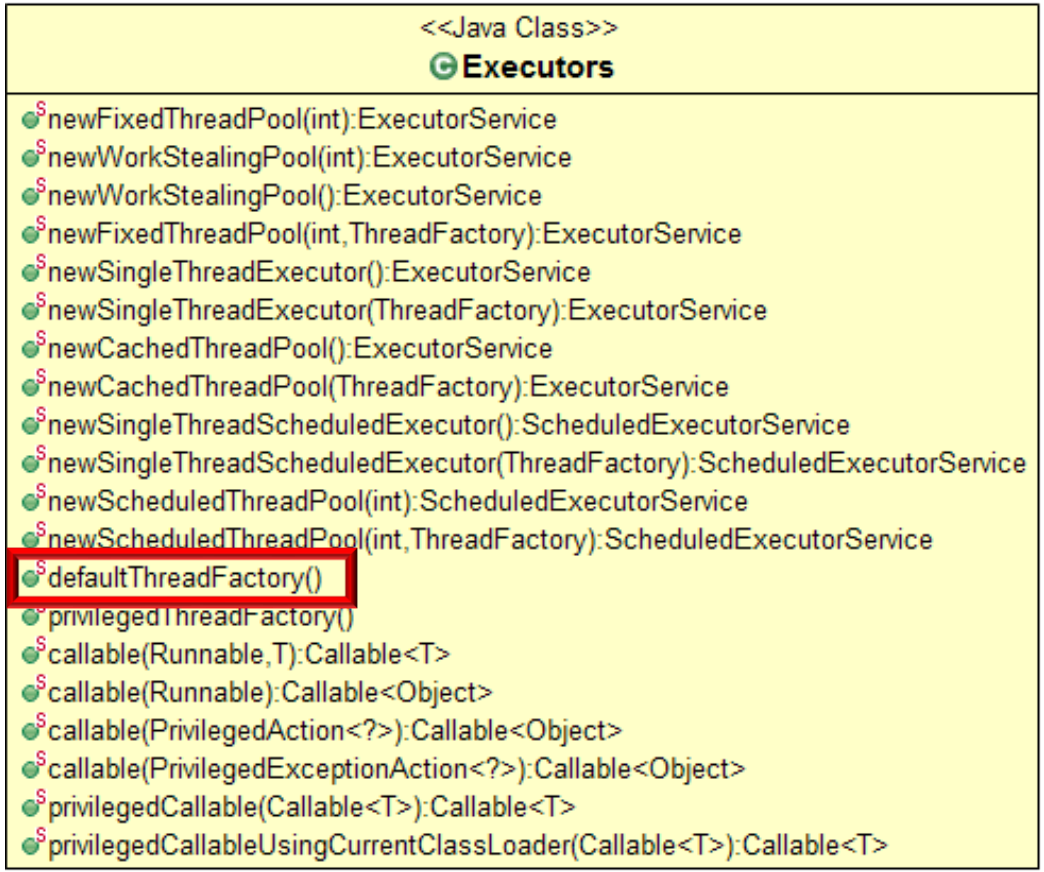
<<Java Class>>
Executors

- newFixedThreadPool(int):ExecutorService
- newWorkStealingPool(int):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>

e.g., cached, fixed, work-stealing thread pools, etc.

The Java Executors Class

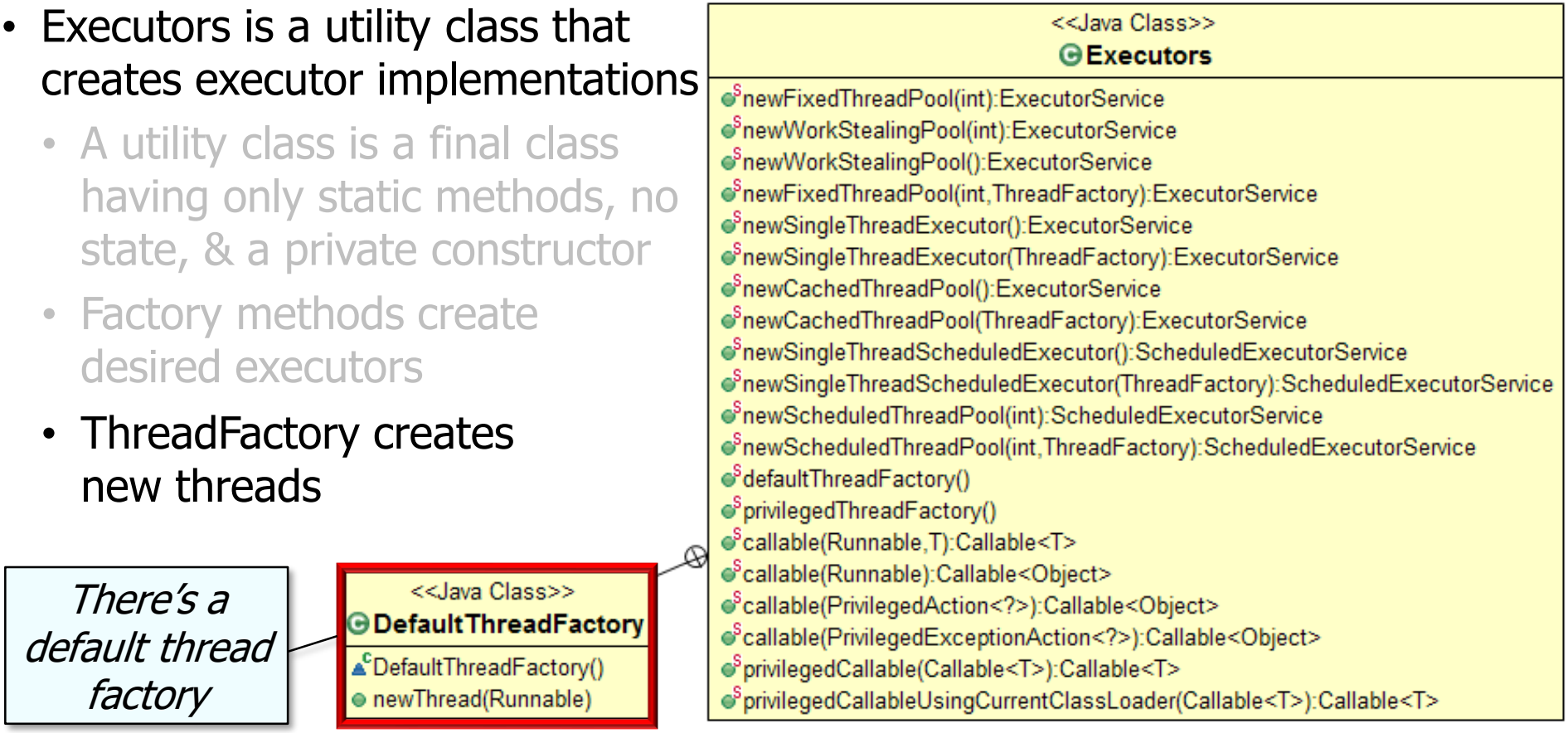
- Executors is a utility class that creates executor implementations
  - A utility class is a final class having only static methods, no state, & a private constructor
  - Factory methods create desired executors
  - ThreadFactory creates new threads



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 having only static methods, no state, & a private constructor
  - Factory methods create desired executors
  - ThreadFactory creates new threads



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

End of Overview of the Java Executors Framework (Part 3)