Subclasses of the Java ForkJoinTask Class

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 how the Java fork-join framework processes tasks in parallel
- Recognize the structure & functionality of the fork-join framework
The Subclasses of ForkJoinTask
The Subclasses of ForkJoinTask

- Programs don’t implement ForkJoinTask...
The Subclasses of ForkJoinTask

- Programs don’t implement ForkJoinTask... but instead extend a subclass & override its compute() hook method

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/package-tree.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/package-tree.html)
The Subclasses of ForkJoinTask

- Programs don’t implement ForkJoinTask… but instead extend a subclass & override its compute() hook method, e.g.
  - **RecursiveAction**
    - Use for computations that do not return results

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/RecursiveAction.html](http://docs.oracle.com/javase/8/docs/api/java/util/concurrent/RecursiveAction.html)
The Subclasses of ForkJoinTask

- Programs don’t implement ForkJoinTask... but instead extend a subclass & override its compute() hook method, e.g.
  - RecursiveAction
  - RecursiveTask
    - Use for computations that do return results

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/RecursiveTask.html
The Subclasses of ForkJoinTask

- Programs don’t implement ForkJoinTask... but instead extend a subclass & override its compute() hook method, e.g.
  - RecursiveAction
  - RecursiveTask
- CountedCompleter
  - Used for computations in which completed actions trigger other actions

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CountedCompleter.html
The Subclasses of ForkJoinTask

- Programs don’t implement ForkJoinTask… but instead extend a subclass & override its compute() hook method, e.g.
  - RecursiveAction
  - RecursiveTask
  - CountedCompleter

These classes aren’t functional interfaces, so they must be subclassed rather than using lambda expressions to implement compute()
The Subclasses of ForkJoinTask

- The parallel streams framework provides a functional facade to ForkJoinTask et al.

See lesson on "Java Parallel Stream Internals: Mapping Onto the Common Fork-Join Pool"
End of Subclasses of the Java ForkJoinTask Class