

# Subclasses of the Java ForkJoinTask Class

**Douglas C. Schmidt**

**[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)**

**[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)**

**Professor of Computer Science**

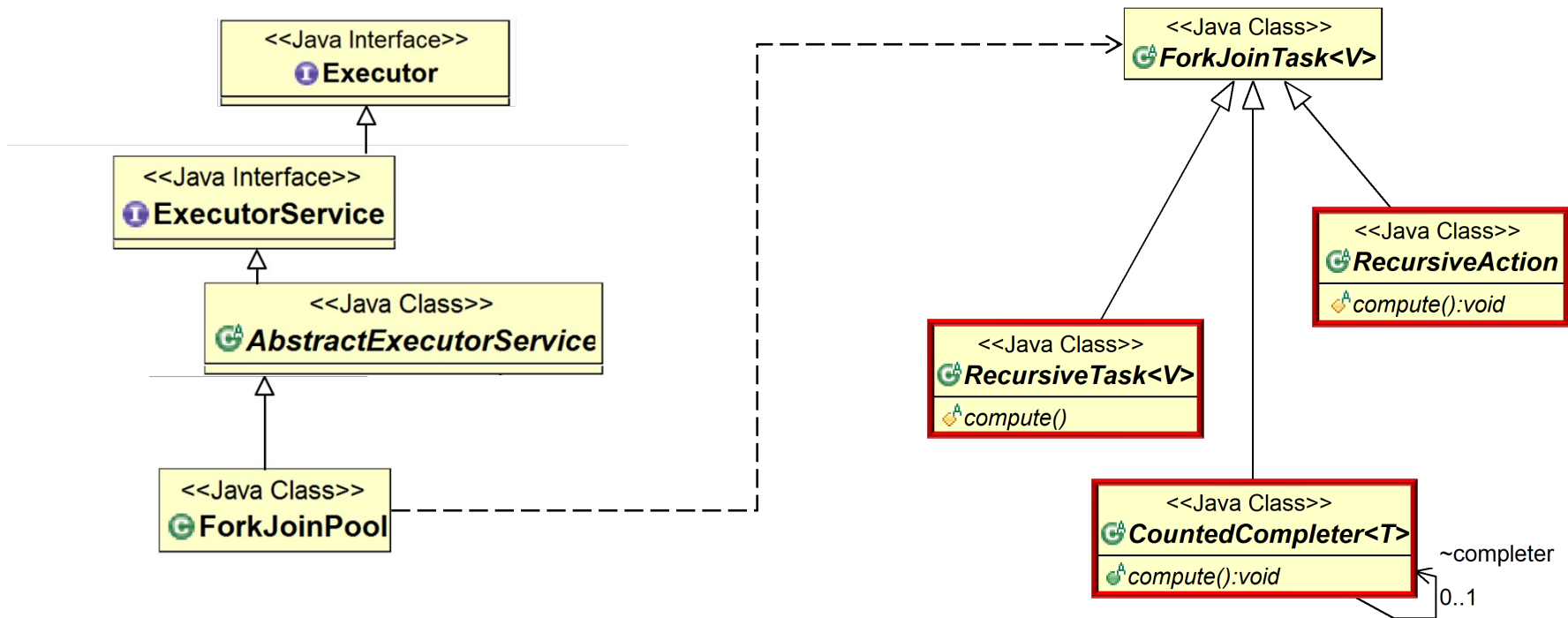
**Institute for Software  
Integrated Systems**

**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

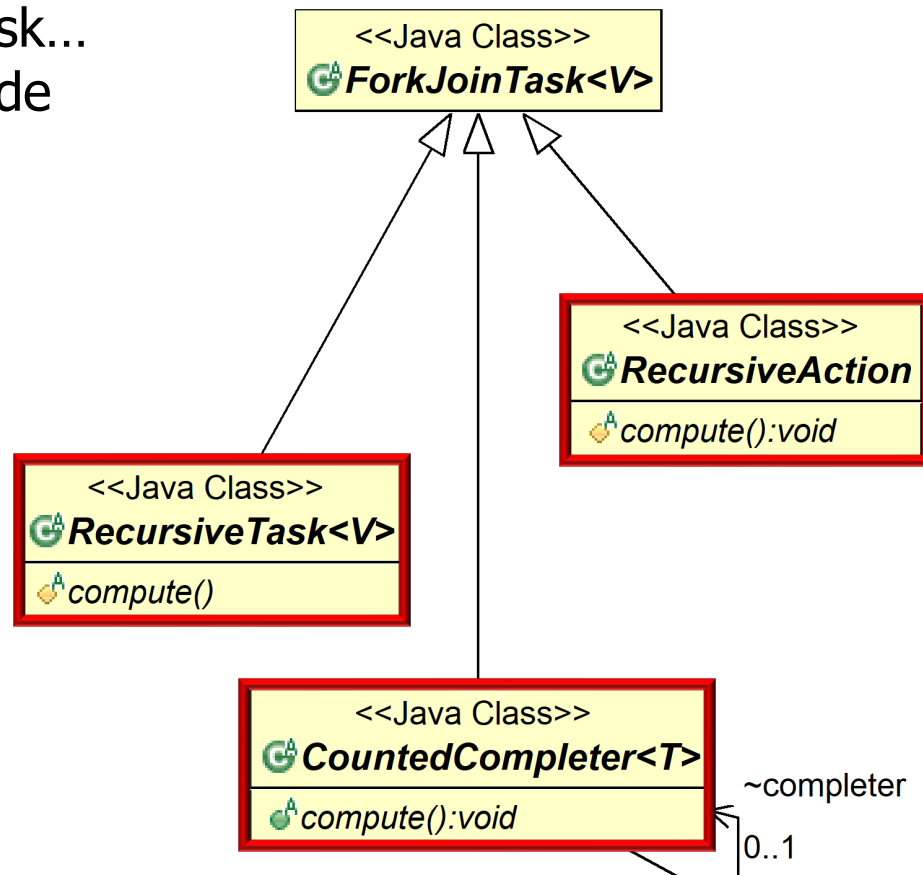
---

- Programs rarely implement ForkJoinTask...



# The Subclasses of ForkJoinTask

- Programs rarely implement ForkJoinTask... but instead extend a subclass & override its compute() hook method

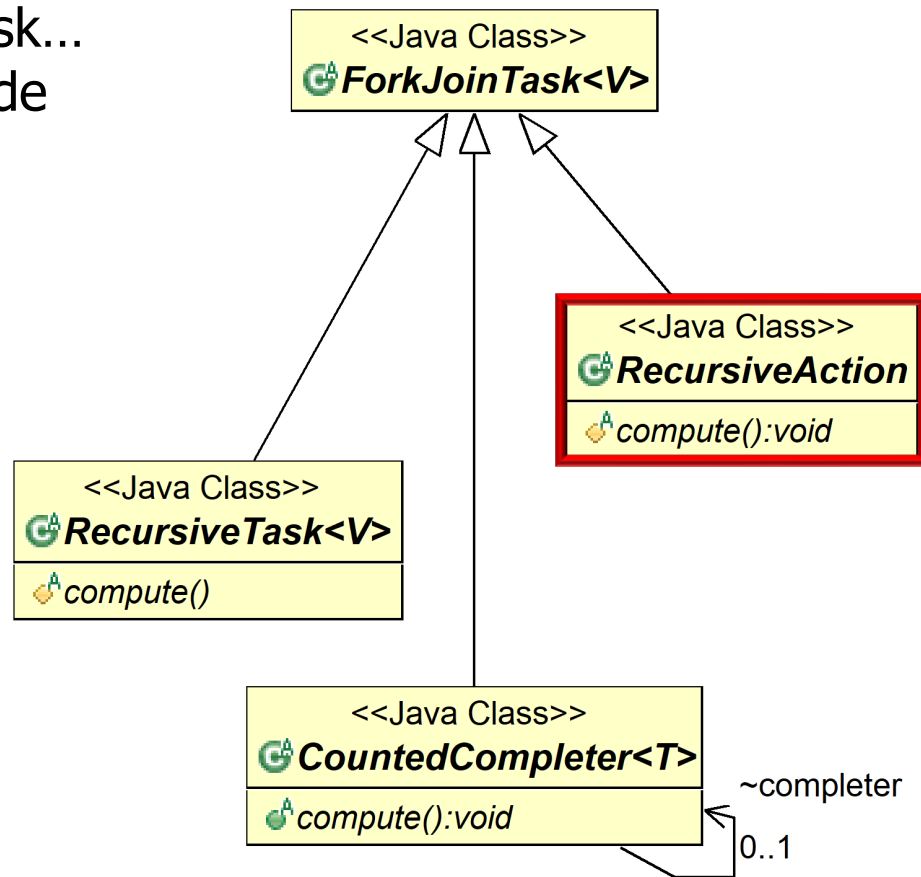


# The Subclasses of ForkJoinTask

- Programs rarely implement ForkJoinTask... but instead extend a subclass & override its compute() hook method, e.g.

- RecursiveAction**

- Use for computations that do not return results

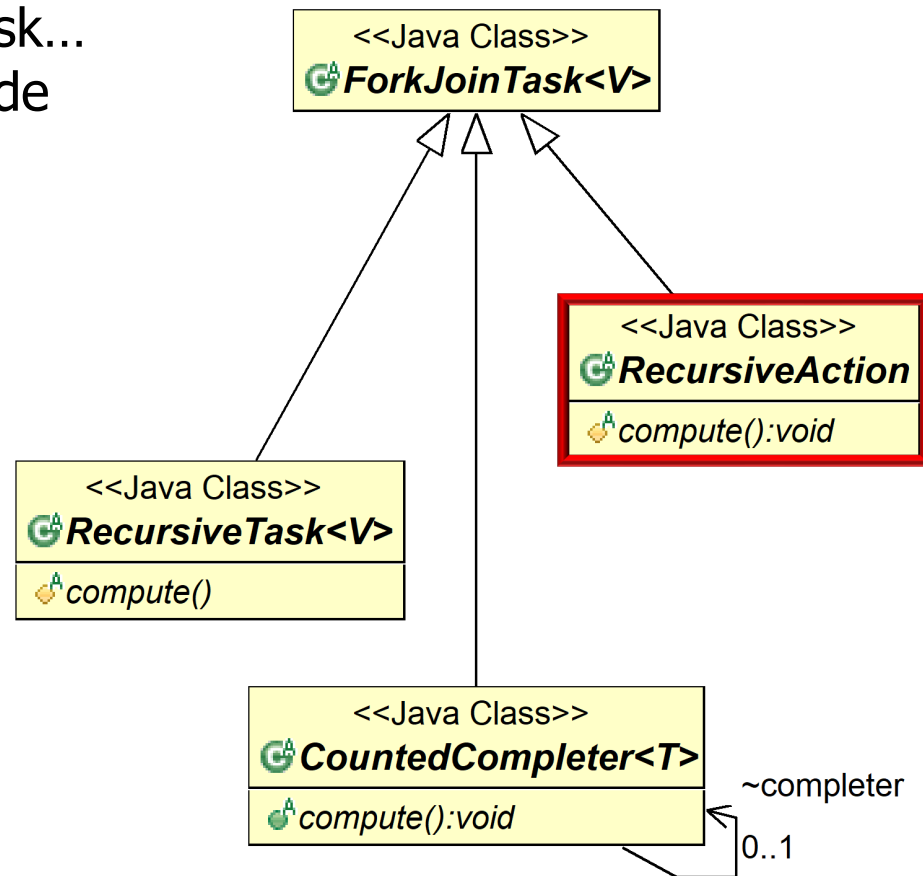


# The Subclasses of ForkJoinTask

- Programs rarely implement ForkJoinTask... but instead extend a subclass & override its compute() hook method, e.g.

- RecursiveAction**

- Use for computations that do not return results
- Ideal for scenarios that perform actions like data modifications, I/O operations, or any other side-effecting operations that return no result

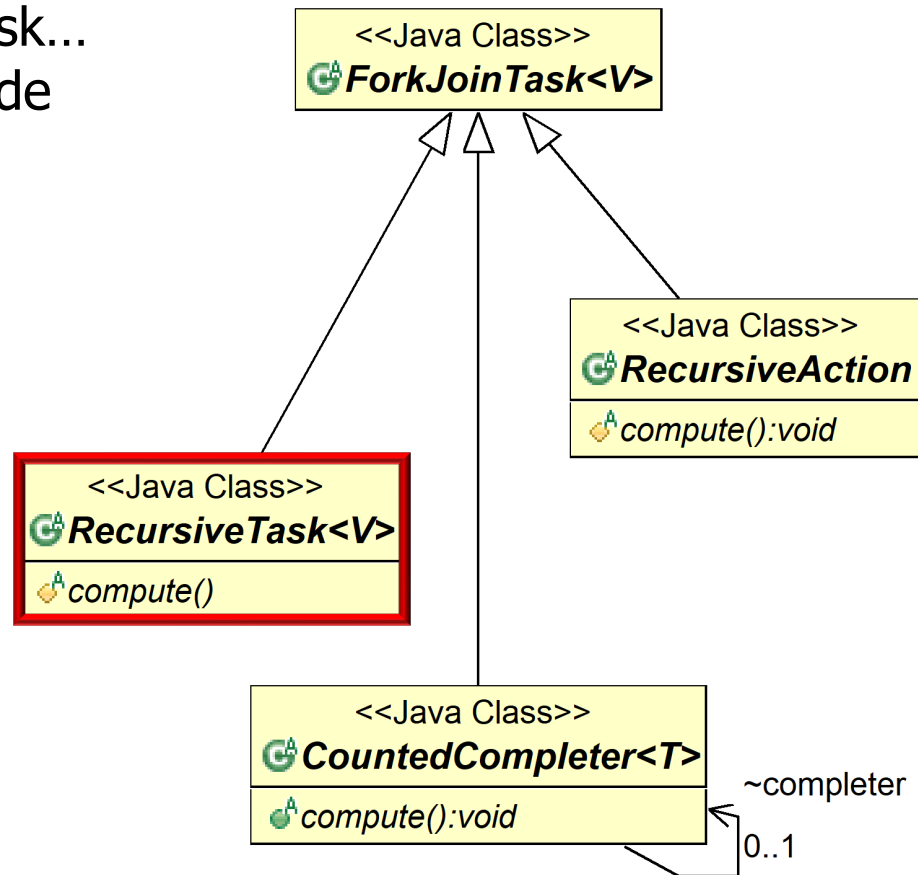


# The Subclasses of ForkJoinTask

- Programs rarely implement ForkJoinTask... but instead extend a subclass & override its compute() hook method, e.g.

- RecursiveTask**

- Use for computations that *do* return results



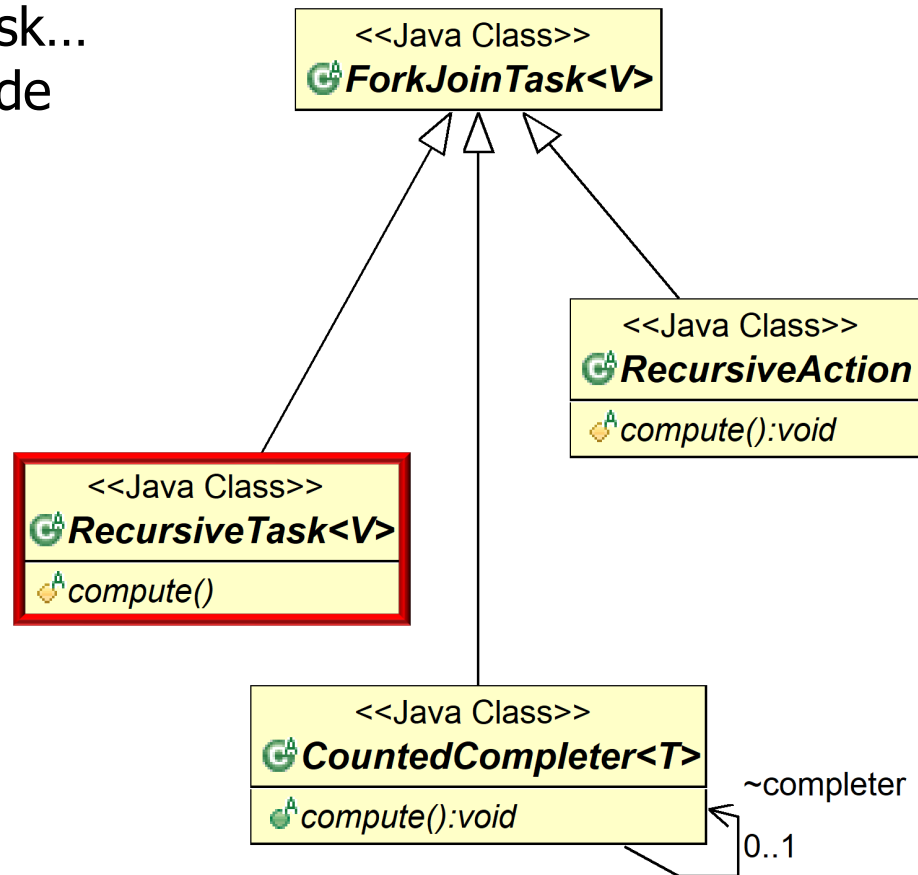


# The Subclasses of ForkJoinTask

- Programs rarely implement ForkJoinTask... but instead extend a subclass & override its compute() hook method, e.g.

- RecursiveTask**

- Use for computations that *do* return results
- Ideal for scenarios that need to process data in parallel & gather results

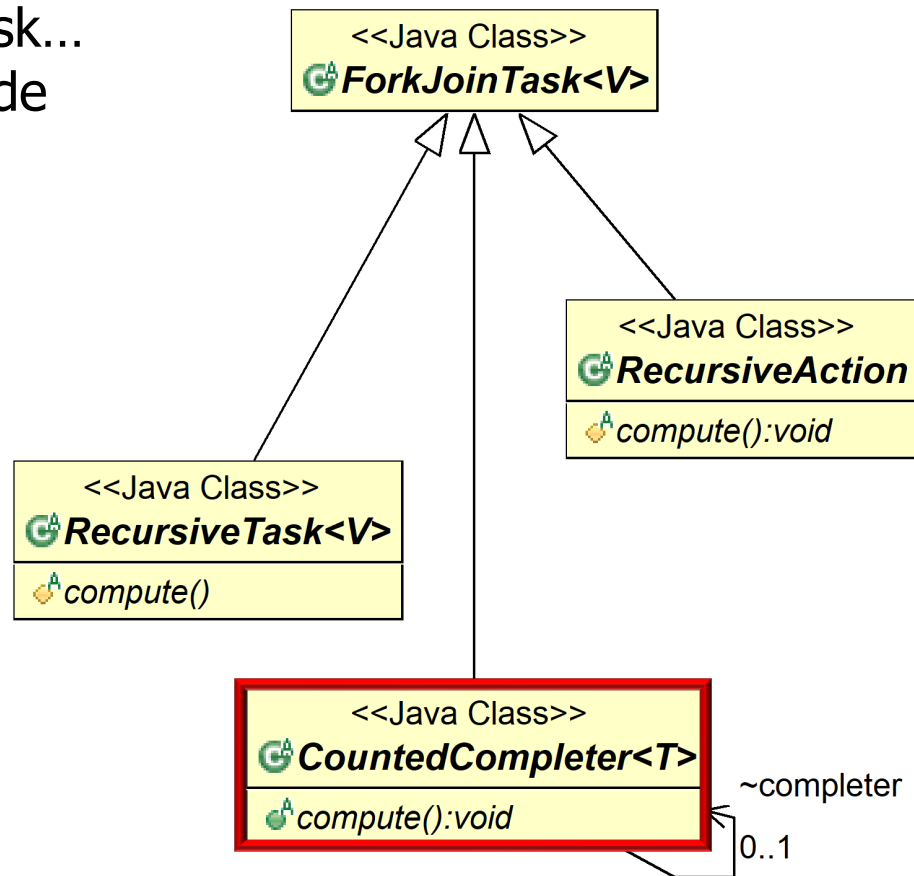


# The Subclasses of ForkJoinTask

- Programs rarely implement ForkJoinTask... but instead extend a subclass & override its compute() hook method, e.g.

- CountedCompleter**

- Used for computations in which completed actions trigger other actions

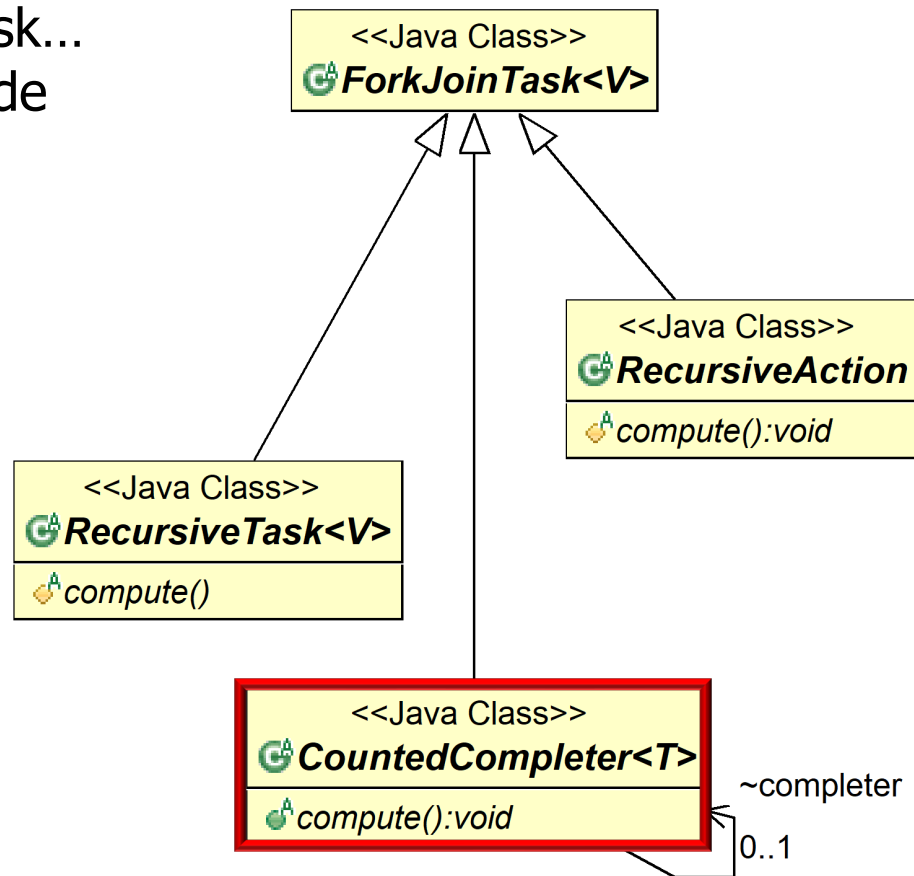


# The Subclasses of ForkJoinTask

- Programs rarely implement ForkJoinTask... but instead extend a subclass & override its compute() hook method, e.g.

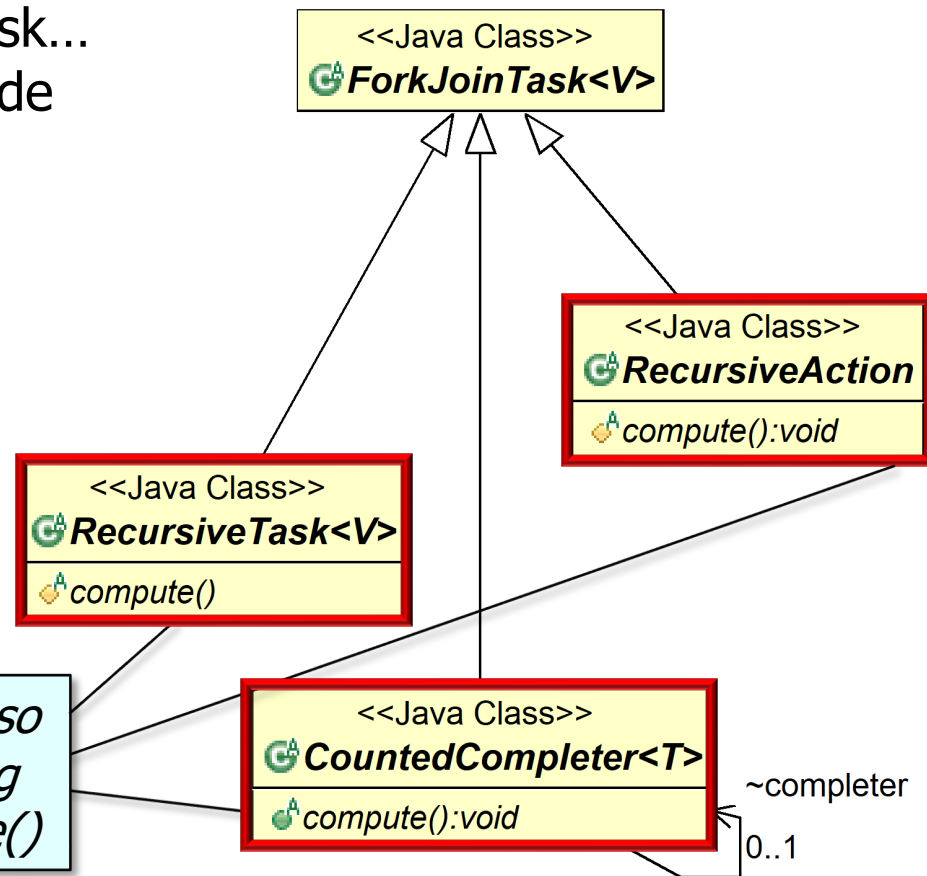
- CountedCompleter**

- Used for computations in which completed actions trigger other actions
- Idea for scenarios when tasks might generate an unpredictable # of sub-tasks & some action must be performed once the task & all its sub-tasks complete



# The Subclasses of ForkJoinTask

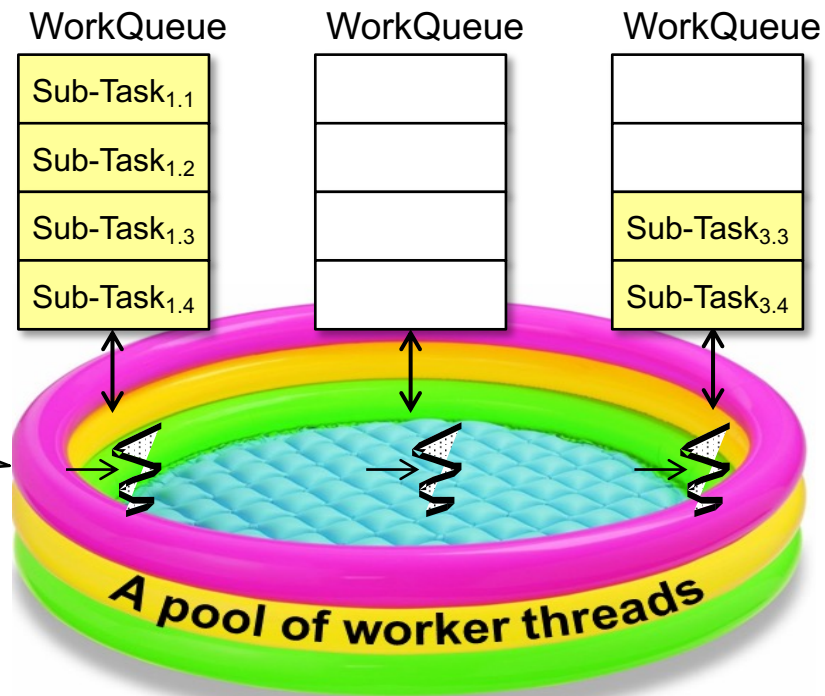
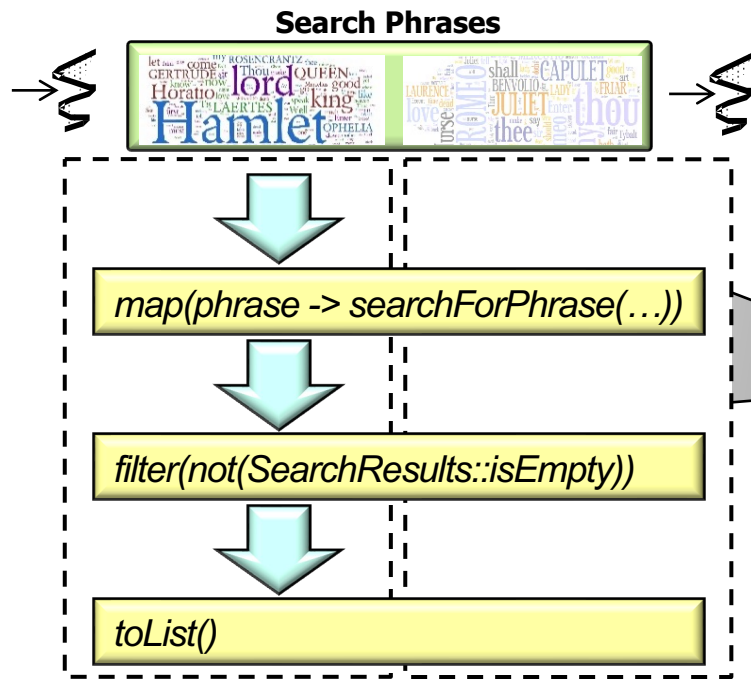
- Programs rarely implement ForkJoinTask... but instead extend a subclass & override its compute() hook method



*These classes aren't functional interfaces, so they must be subclassed rather than using lambda expressions to implement compute()*

# The Subclasses of ForkJoinTask

- The Java parallel streams framework provides a functional facade to ForkJoinTask-related classes



See lesson on "Java Parallel Stream Internals: Mapping Onto the Common Fork-Join Pool"

---

# End of Subclasses of the Java ForkJoinTask Class