Understanding Advanced Java Completable Future Features: Grouping Completion Stage Methods

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

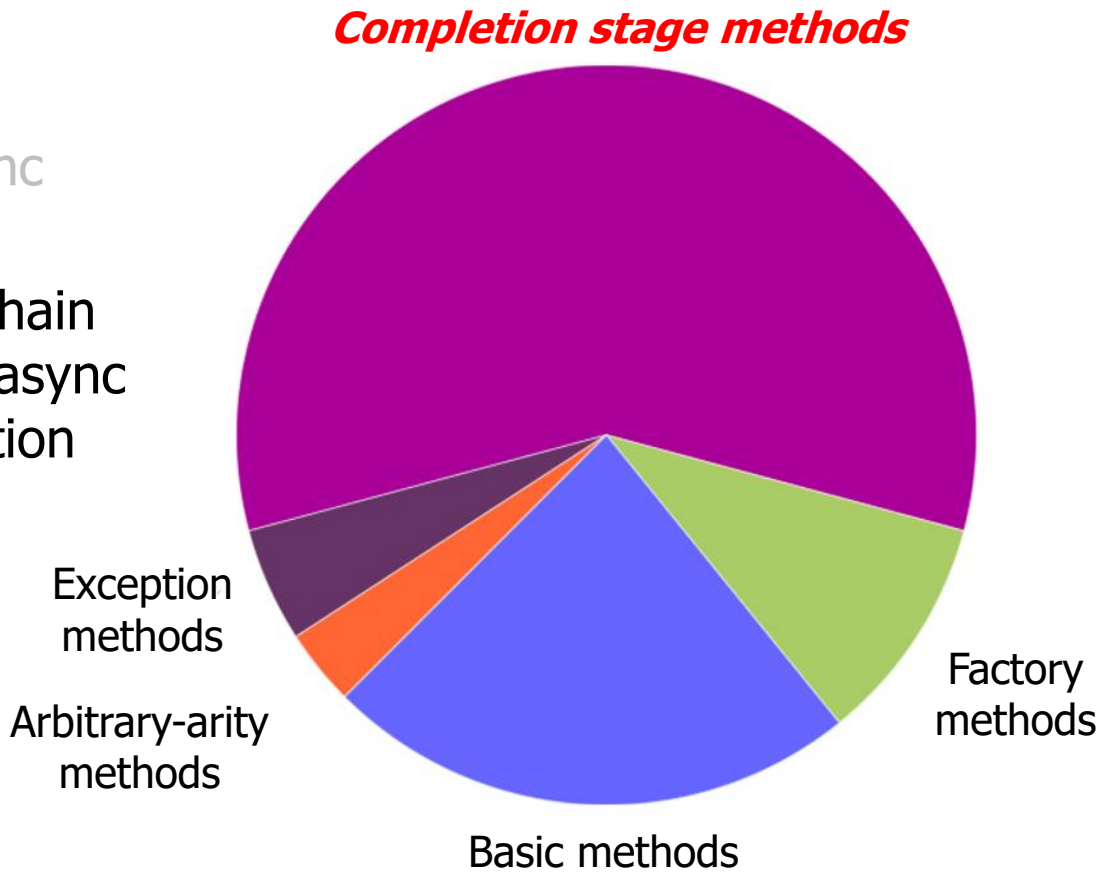
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

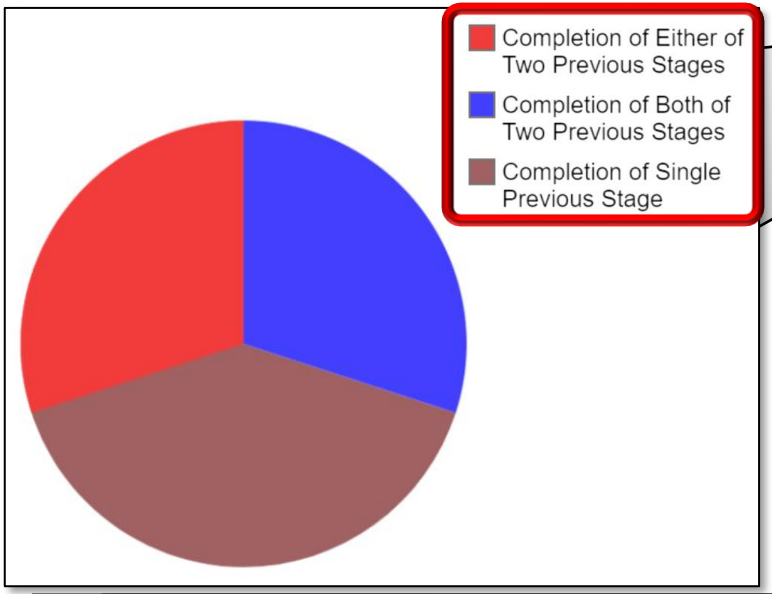
- Understand advanced features of completable futures, e.g.
  - Factory methods initiate async computations
  - Completion stage methods chain together actions to perform async result processing & composition
    - Method grouping

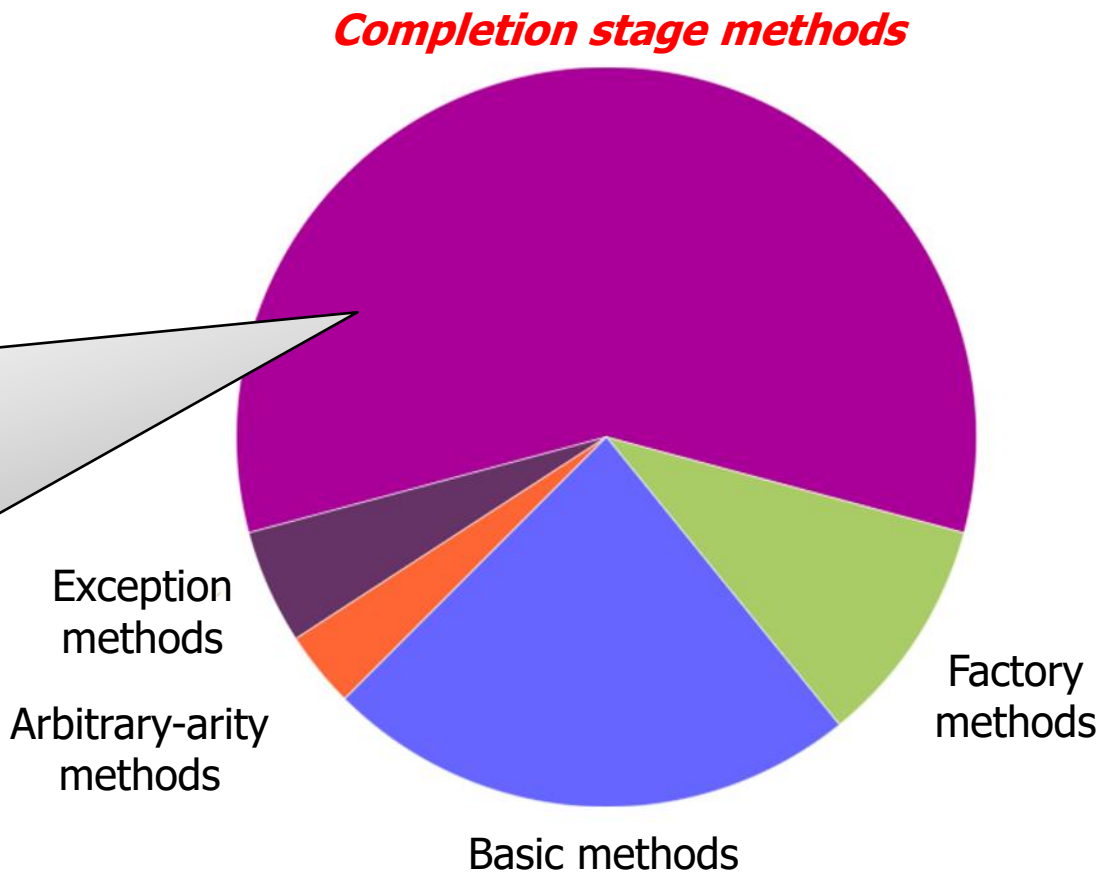


Grouping CompletableFuture Completion Stage Methods

Grouping CompletableFuture Completion Stage Methods

- Completion stage methods are grouped based on how a stage is triggered by one or more previous stage(s)





See www.jesperdj.com/2015/09/26/the-future-is-completable-in-java-8

Grouping CompletableFuture Completion Stage Methods

- Completion stage methods are grouped based on how a stage is triggered by one or more previous stage(s)
  - Completion of a single previous stage

These methods run in the invoking thread or the same thread as previous stage

Methods	Params	Returns	Behavior
thenApply (Async)	Function	CompletableFuture with Function result	Apply function to result of the previous stage
thenCompose (Async)	Function	CompletableFuture with Function result directly, *not* a nested future	Apply function to result of the previous stage
thenAccept (Async)	Consumer	CompletableFuture<Void>	Consumer handles result of previous stage
thenRun (Async)	Runnable	CompletableFuture<Void>	Run action w/out returning value

The thread that executes these methods depends on various runtime factors

Grouping CompletableFuture Completion Stage Methods

- Completion stage methods are grouped based on how a stage is triggered by one or more previous stage(s)
  - Completion of a single previous stage

**Async() variants run in some thread pool*

Methods	Params	Returns	Behavior
thenApply (Async)	Function	CompletableFuture with Function result	Apply function to result of the previous stage
then Compose (Async)	Function	CompletableFuture with Function result directly, *not* a nested future	Apply function to result of the previous stage
then Accept (Async)	Consumer	CompletableFuture<Void>	Consumer handles result of previous stage
thenRun (Async)	Runnable	CompletableFuture<Void>	Run action w/out returning value

See blog.krecan.net/2013/12/25/completablefutures-why-to-use-async-methods

Grouping CompletableFuture Completion Stage Methods

- Completion stage methods are grouped based on how a stage is triggered by one or more previous stage(s)
  - Completion of a single previous stage
  - Completion of both of two previous stages
    - i.e., an "and"

Methods	Params	Returns	Behavior
then Combine (Async)	Bi Function	Completable Future with Bi Function result	Apply bifunction to results of both previous stages
then Accept Both (Async)	Bi Consumer	Completable Future<Void>	BiConsumer handles results of both previous stages
runAfter Both (Async)	Runnable	Completable Future<Void>	Run action when both previous stages complete

Grouping CompletableFuture Completion Stage Methods

- Completion stage methods are grouped based on how a stage is triggered by one or more previous stage(s)
  - Completion of a single previous stage
  - Completion of both of two previous stages
  - Completion of either of two previous stages
    - i.e., an "or"

Methods	Params	Returns	Behavior
applyToEither (Async)	Function	CompletableFuture with Function result	Apply function to results of either previous stage
acceptEither (Async)	Consumer	CompletableFuture<Void>	Consumer handles results of either previous stage
runAfterEither (Async)	Runnable	CompletableFuture<Void>	Run action when either previous stage completes

End of Understand Advanced Java Completable Future Features: Grouping Completion Stage Methods