The Image Counter Case Study (Part 2)

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

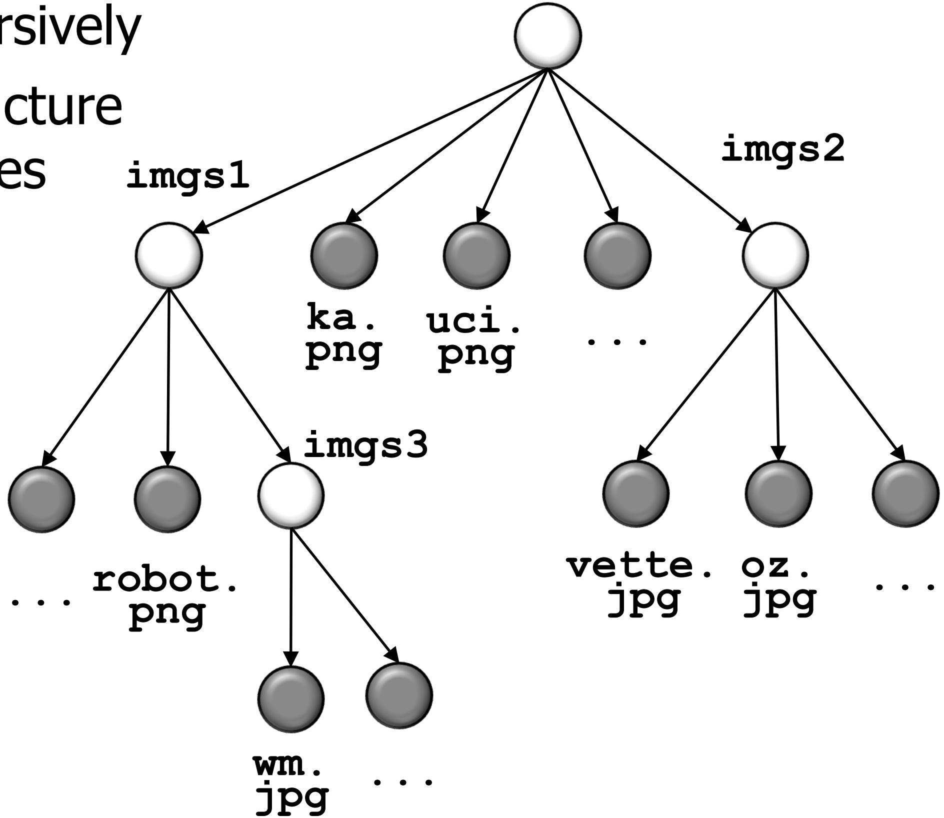
Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

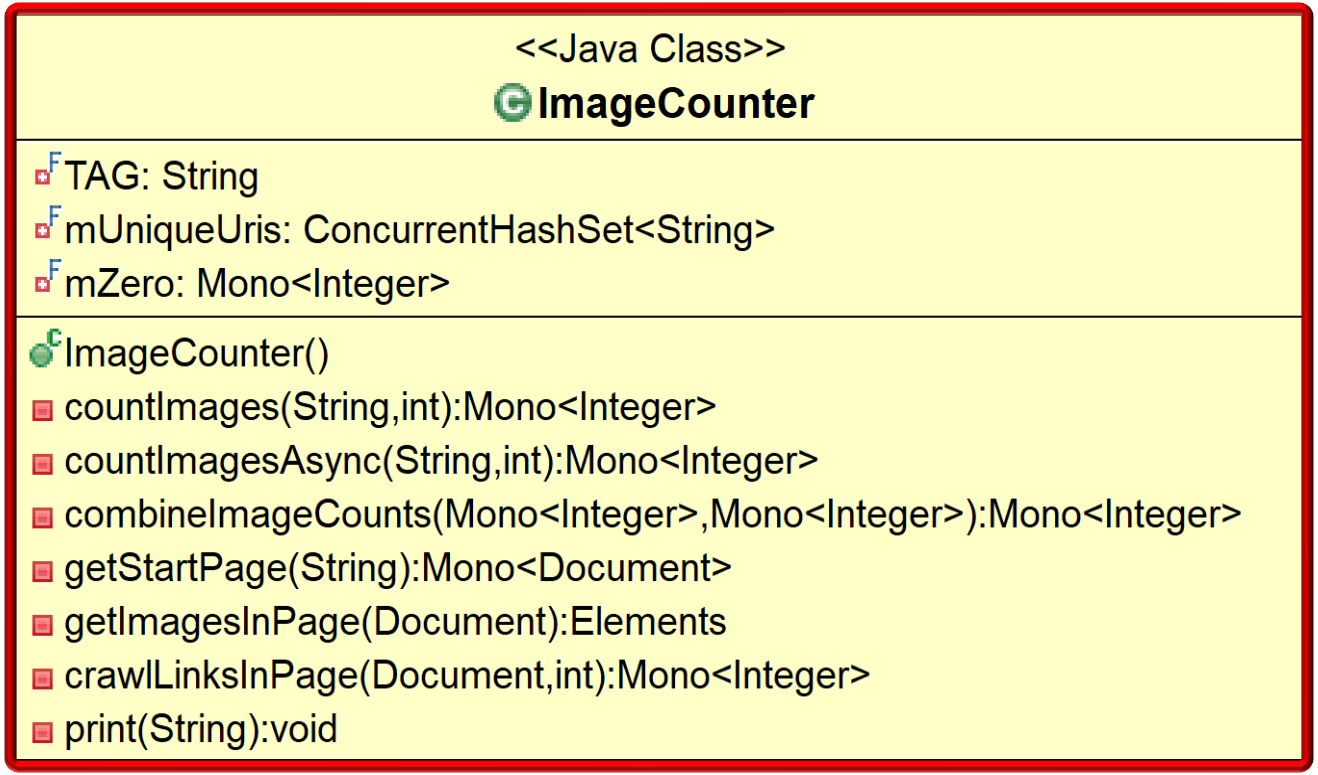
- Know how Mono, Flux, ParallelFlux, & Scheduler operators in Project Reactor are applied to crawl web pages recursively
  - i.e., crawl a hierarchical folder structure recursively to count the # of images from a root URI



See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/ImageCounter

Learning Objectives in this Part of the Lesson

- Know how Mono, Flux, ParallelFlux, & Scheduler operators in Project Reactor are applied to crawl web pages recursively
- Understand the ImageCounter class implementation



<<Java Class>>

ImageCounter

TAG: String

mUniqueUris: ConcurrentHashSet<String>

mZero: Mono<Integer>

ImageCounter()

countImages(String,int):Mono<Integer>

countImagesAsync(String,int):Mono<Integer>

combineImageCounts(Mono<Integer>,Mono<Integer>):Mono<Integer>

getStartPage(String):Mono<Document>

getImagesInPage(Document):Elements

crawlLinksInPage(Document,int):Mono<Integer>

print(String):void

See Reactive/ImageCounter/src/main/java/ImageCounter.java

End of the Image Counter Case Study (Part 2)