Understand the Java Parallel Image
StreamGang Structure & Functionality

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 structure & functionality of the ImageStreamGang app
- It applies several Java parallelism frameworks
- Focus is on integrating object-oriented & functional programming paradigms

This design shows the synergy between object-oriented & functional programming
Learning Objectives in this Part of the Lesson

• Understand the structure & functionality of the ImageStreamGang app
• It applies several Java parallelism frameworks
• Focus is on integrating object-oriented & functional programming paradigms

Patterns are used to emphasize key roles & responsibilities in the app’s design
The Structure of the ImageStreamGang App
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

See en.wikipedia.org/wiki/Unified_Modeling_Language
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

These classes apply Java features to image downloading & processing
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

A framework for initiating streams that process input from a list of elements
Customizes the StreamGang framework to download & process images ...

The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

```
<<Java Class>>
 MainActivityBase

<<Java Class>>
 MainActivity

<<Java Class>>
 ImageStreamGang

<<Java Class>>
 ImageStreamSequential

<<Java Class>>
 ResultsActivity

<<Java Class>>
 StreamGang<E>

<<Java Class>>
 Image

<<Java Class>>
 Filter

<<Java Class>>
 ImageStreamCompletableFutureBase

<<Java Class>>
 ImageStreamParallel

<<Java Class>>
 ImageAdapter

<<Java Class>>
 FilterDecorator

<<Java Class>>
 NullFilter

<<Java Class>>
 FilterDecoratorWithImage

<<Java Class>>
 GrayScaleFilter

<<Java Class>>
 ImageStreamCompletableFuture1

<<Java Class>>
 ImageStreamCompletableFuture2
```
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

... based on different Java concurrency & parallelism frameworks
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

Uses Java streams to download & filter images sequentially
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

Uses Java parallel streams to download & filter images concurrently
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

Uses Java CompletableFutures to download & filter images asynchronously.
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

Stores image meta-data & provides methods for common image-/file-related tasks
This class hierarchy applies operations to filter & store images
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

Provides the user interface for an Android app
The Structure of the ImageStreamGang App

- UML class diagram for the object-oriented ImageStreamGang app design

There's a Java console version of ImageStreamGang that shares most of the code.
Running the Image StreamGang App
Starting ImageStreamGangTest
Printing 4 results for input file 1 from fastest to slowest
COMPLETABLE_FUTURES_1 executed in 312 msecs
COMPLETABLE_FUTURES_2 executed in 335 msecs
PARALLEL_STREAM executed in 428 msecs
SEQUENTIAL_STREAM executed in 981 msecs

Printing 4 results for input file 2 from fastest to slowest
COMPLETABLE_FUTURES_2 executed in 82 msecs
COMPLETABLE_FUTURES_1 executed in 83 msecs
PARALLEL_STREAM executed in 102 msecs
SEQUENTIAL_STREAM executed in 251 msecs
Ending ImageStreamGangTest

Tests conducted on a 2.6 GHz six-core Lenovo P52 with 64 Gbytes of RAM
End of Understand the Java
Parallel ImageStreamGang Structure & Functionality