

# Understand the Java Parallel Image StreamGang App Structure

**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**

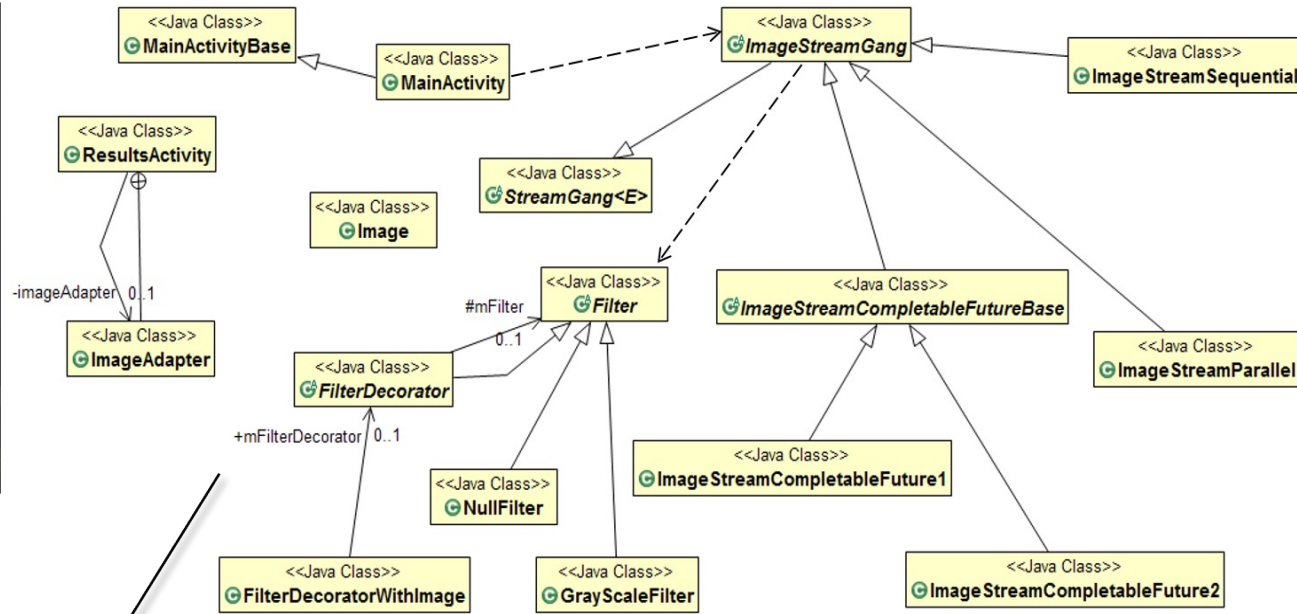
**Institute for Software  
Integrated Systems**

**Vanderbilt University  
Nashville, Tennessee, USA**



# Learning Objectives in this Part of the Lesson

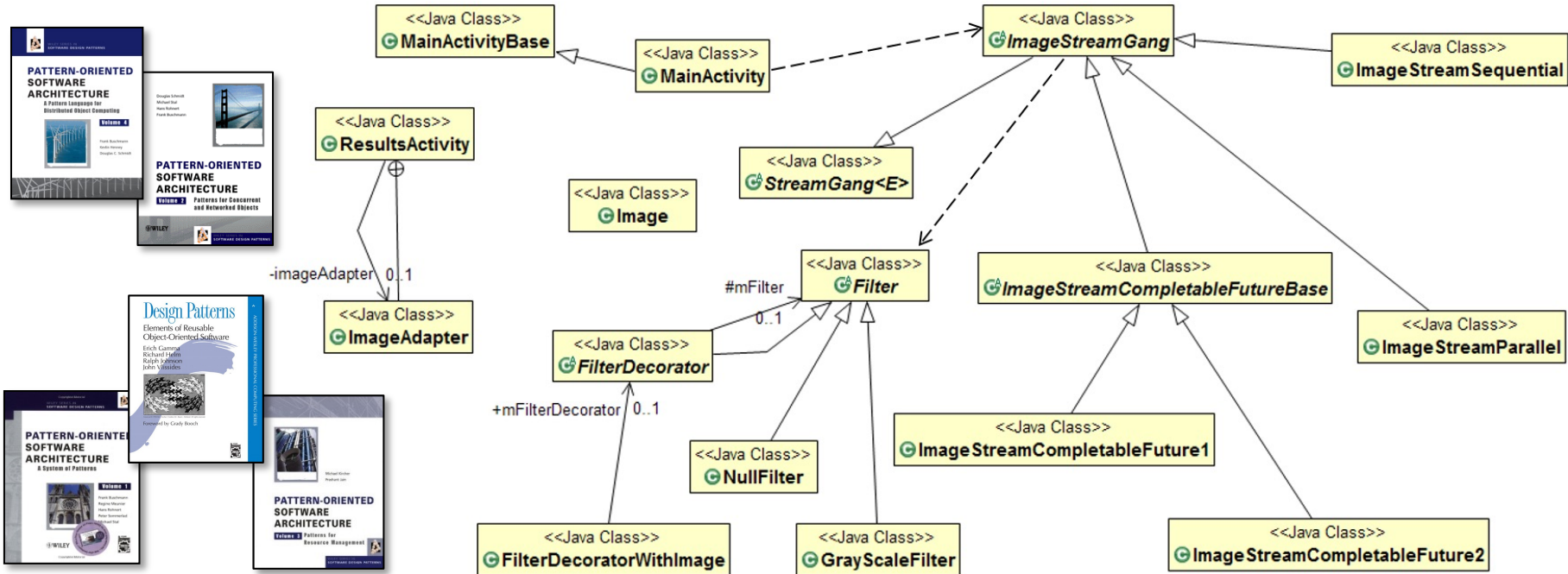
- Understand purpose of the ImageStreamGang app
- Recognize patterns applied in the ImageStreamGang app
- Know how the structure of the ImageStreamGang app



*This design shows the synergy between object-oriented & functional programming*

# Learning Objectives in this Part of the Lesson

- Understand purpose of the ImageStreamGang app
- Recognize patterns applied in the ImageStreamGang app
- Know how the structure of the ImageStreamGang app



Patterns are used to emphasize key roles & responsibilities in the app's design

# Learning Objectives in this Part of the Lesson

- Understand purpose of the ImageStreamGang app
- Recognize patterns applied in the ImageStreamGang app
- Know how the structure of the ImageStreamGang app
  - & its performance for various implementation strategies

Starting ImageStreamGangTest

Printing 4 results for input file 1 from fastest to slowest

COMPLETABLE\_FUTURES\_2 executed in 153 msecs

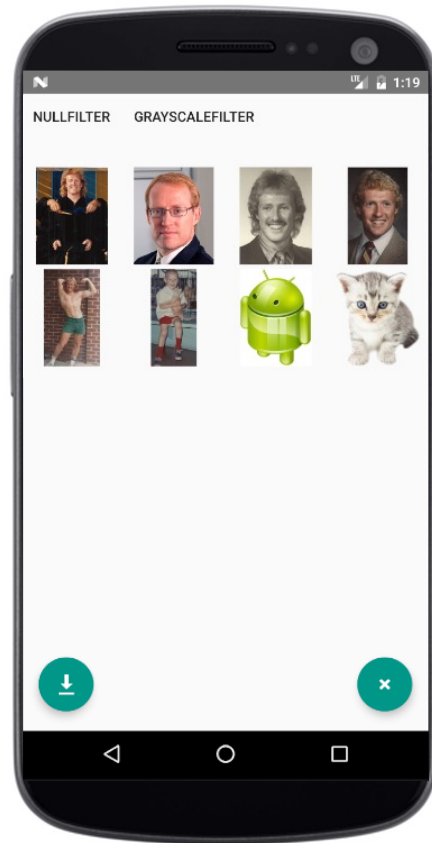
COMPLETABLE\_FUTURES\_1 executed in 251 msecs

PARALLEL\_STREAM executed in 300 msecs

SEQUENTIAL\_STREAM executed in 1026 msecs

...

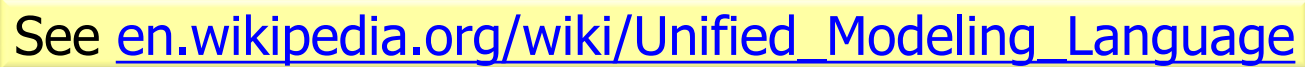
Ending ImageStreamGangTest



---

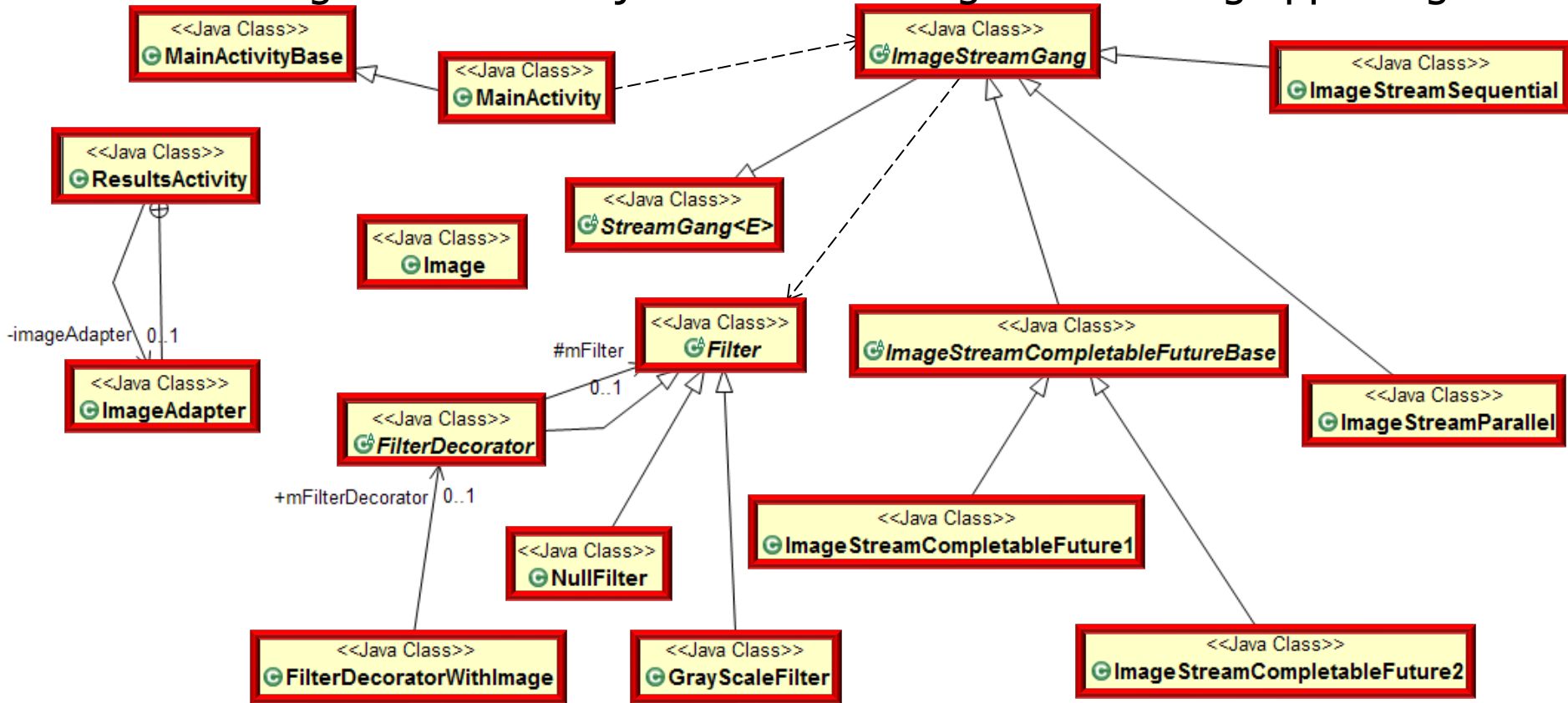
# The Structure of the ImageStreamGang App

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



# The Structure of the ImageStreamGang App

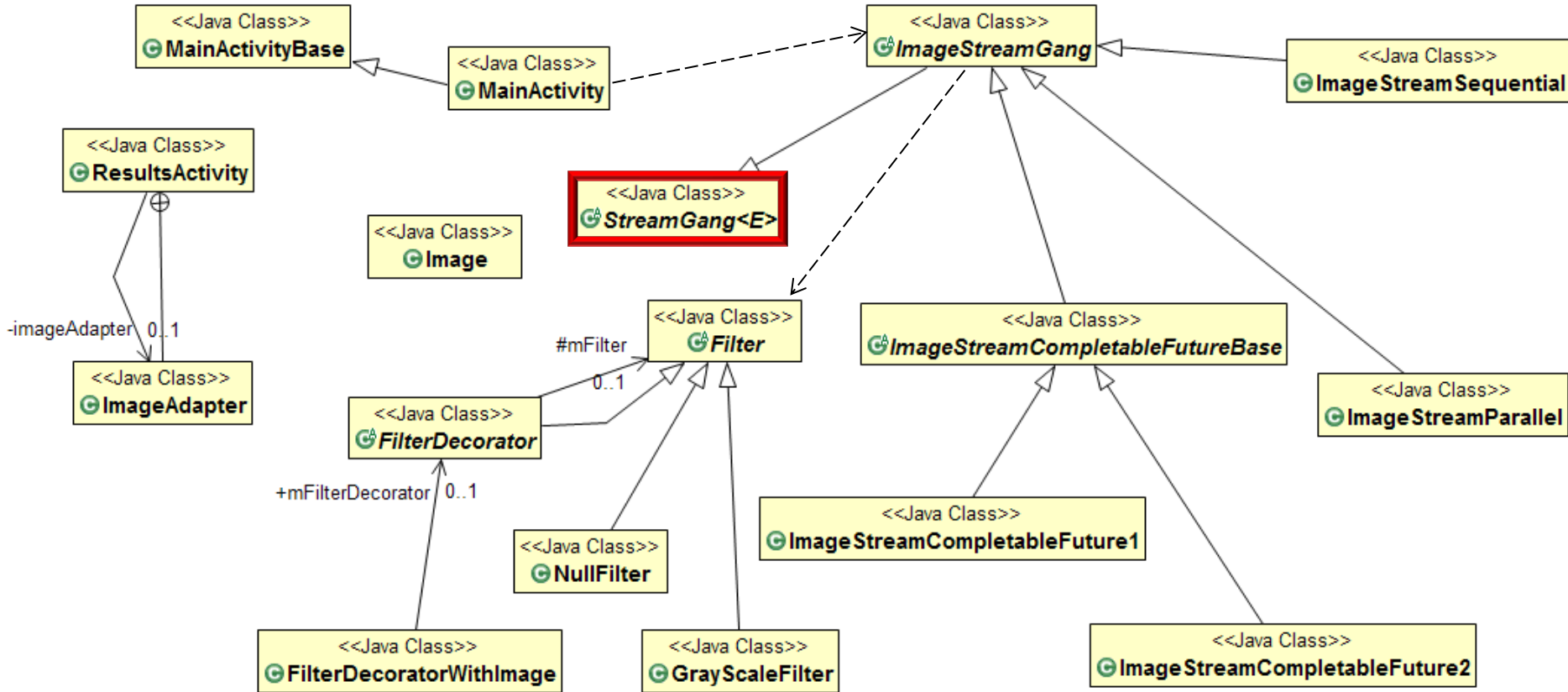
- UML class diagram for the object-oriented ImageStreamGang app design



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

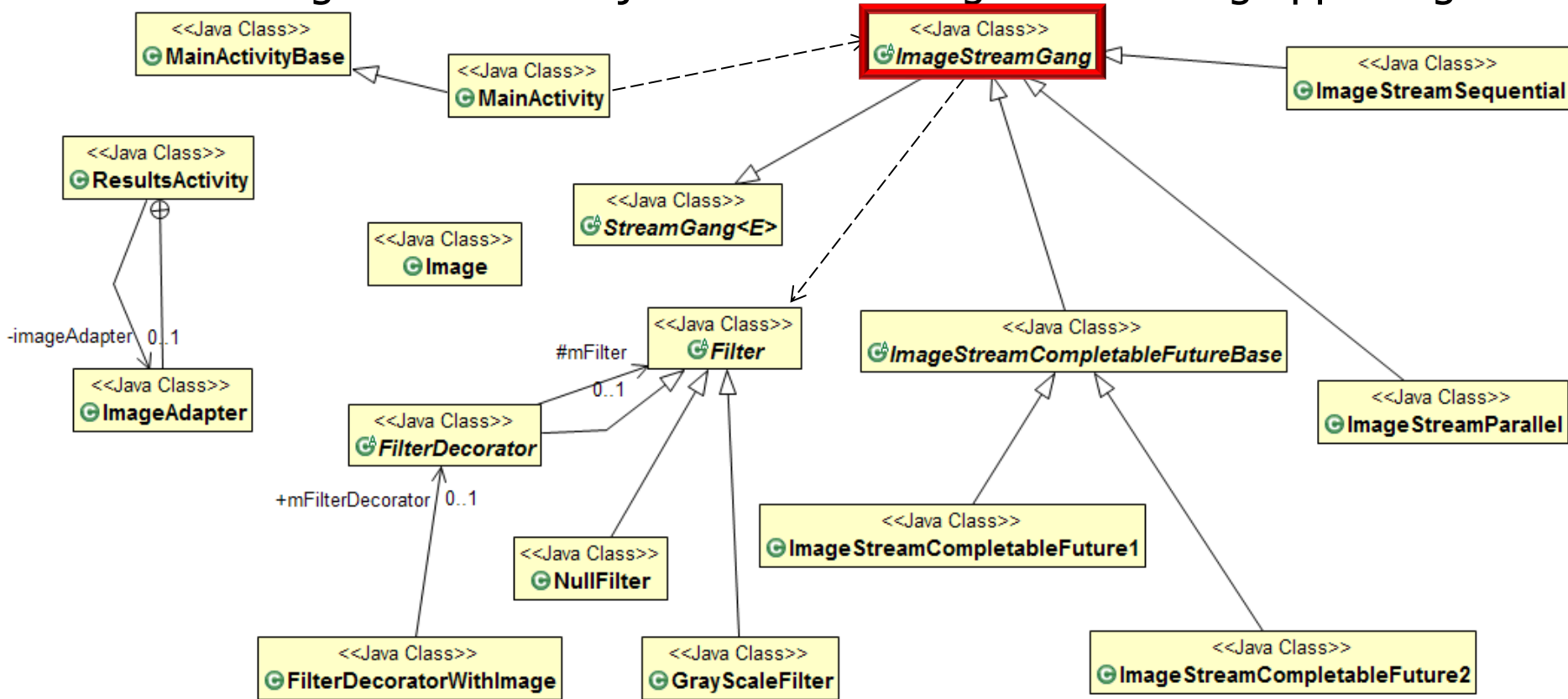


A framework for initiating streams that process input from a list of elements



# The Structure of the ImageStreamGang App

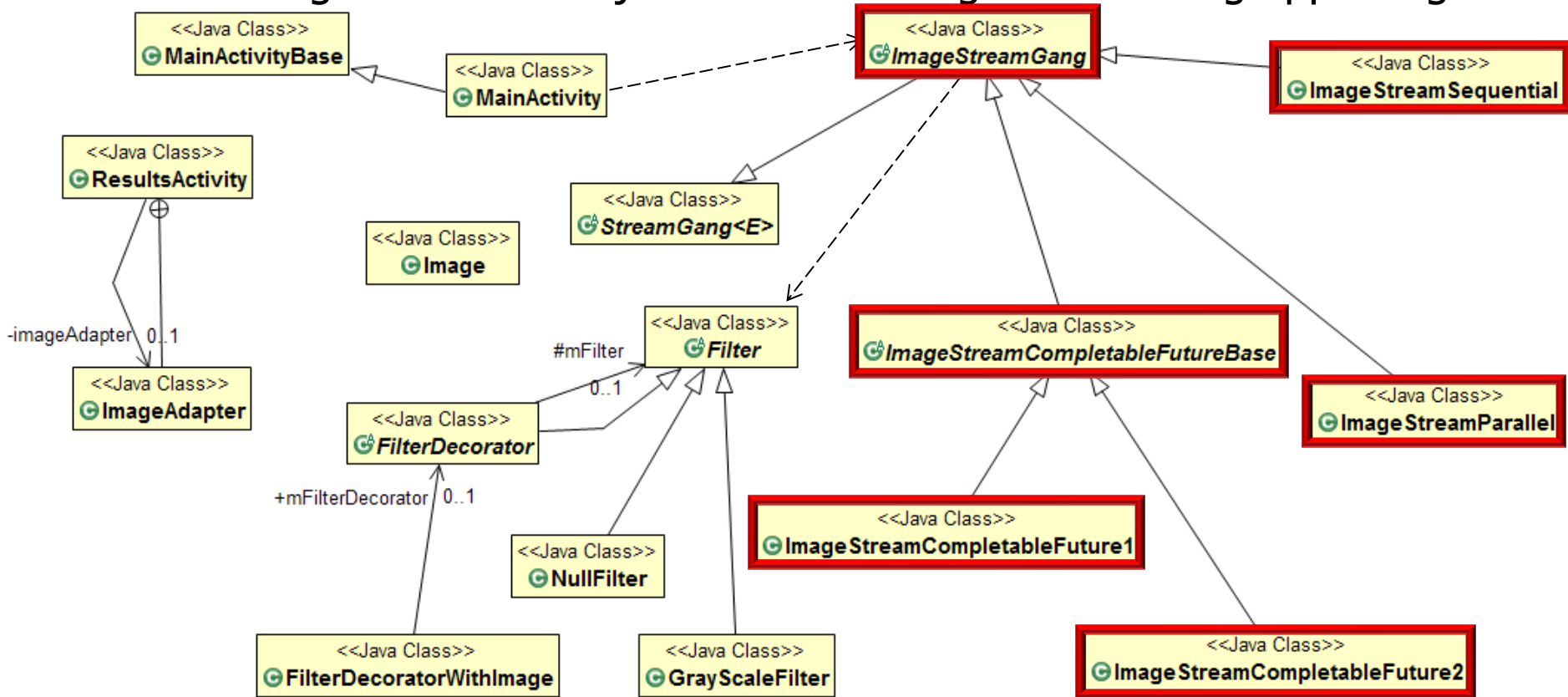
- UML class diagram for the object-oriented ImageStreamGang app design



Customizes the StreamGang framework to download & process images ...

# The Structure of the ImageStreamGang App

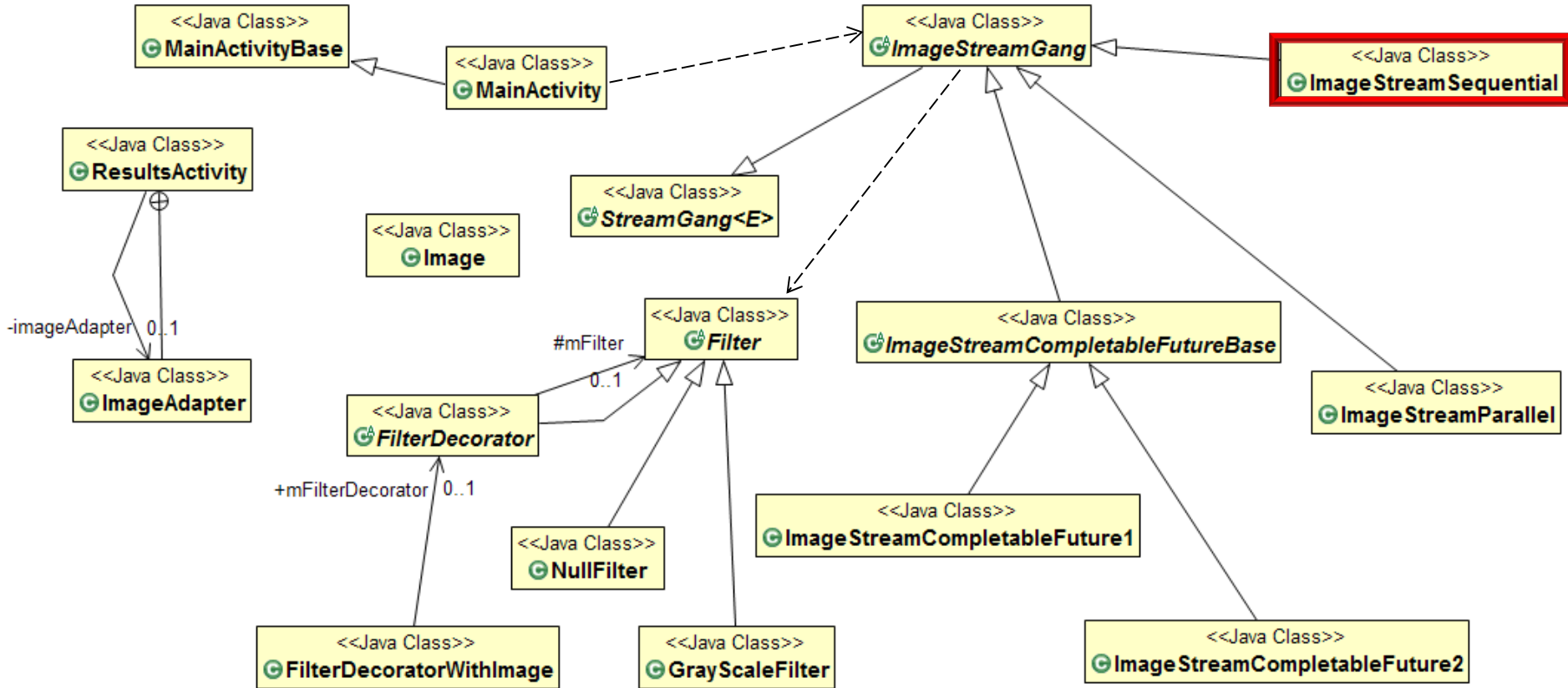
- UML class diagram for the object-oriented ImageStreamGang app design



... based on different Java concurrency & parallelism frameworks

# The Structure of the ImageStreamGang App

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



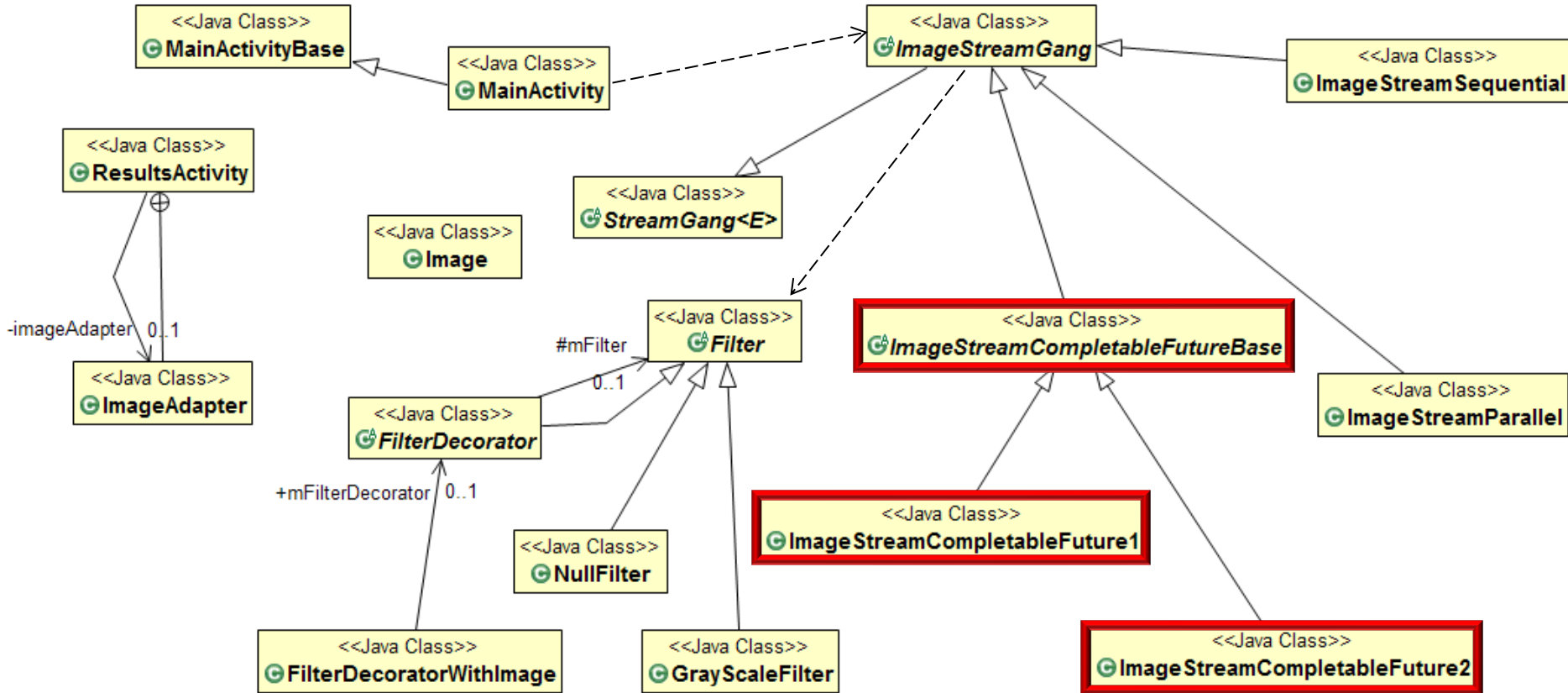
Uses Java streams to download & filter images sequentially

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



# The Structure of the ImageStreamGang App

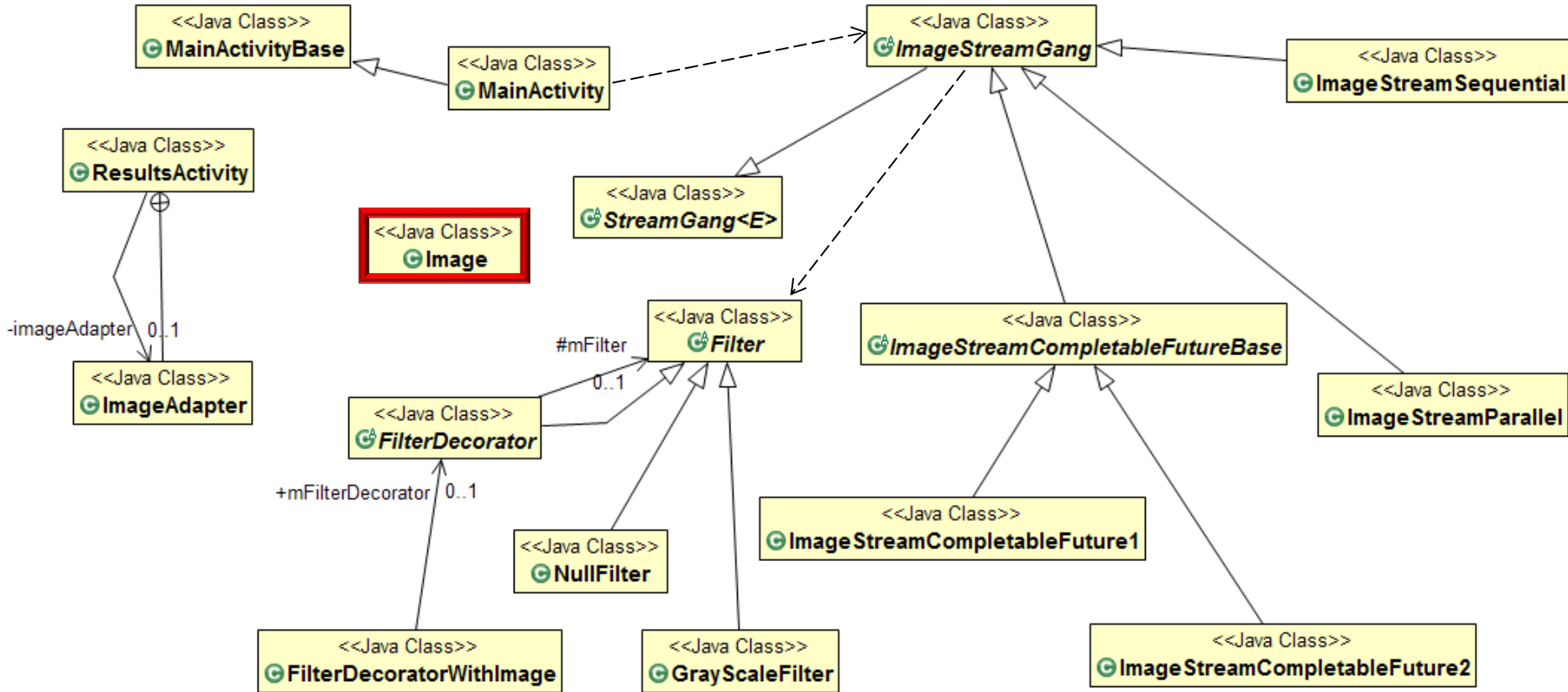
- UML class diagram for the object-oriented ImageStreamGang app design



Uses Java CompletableFuture to download & filter images asynchronously

# The Structure of the ImageStreamGang App

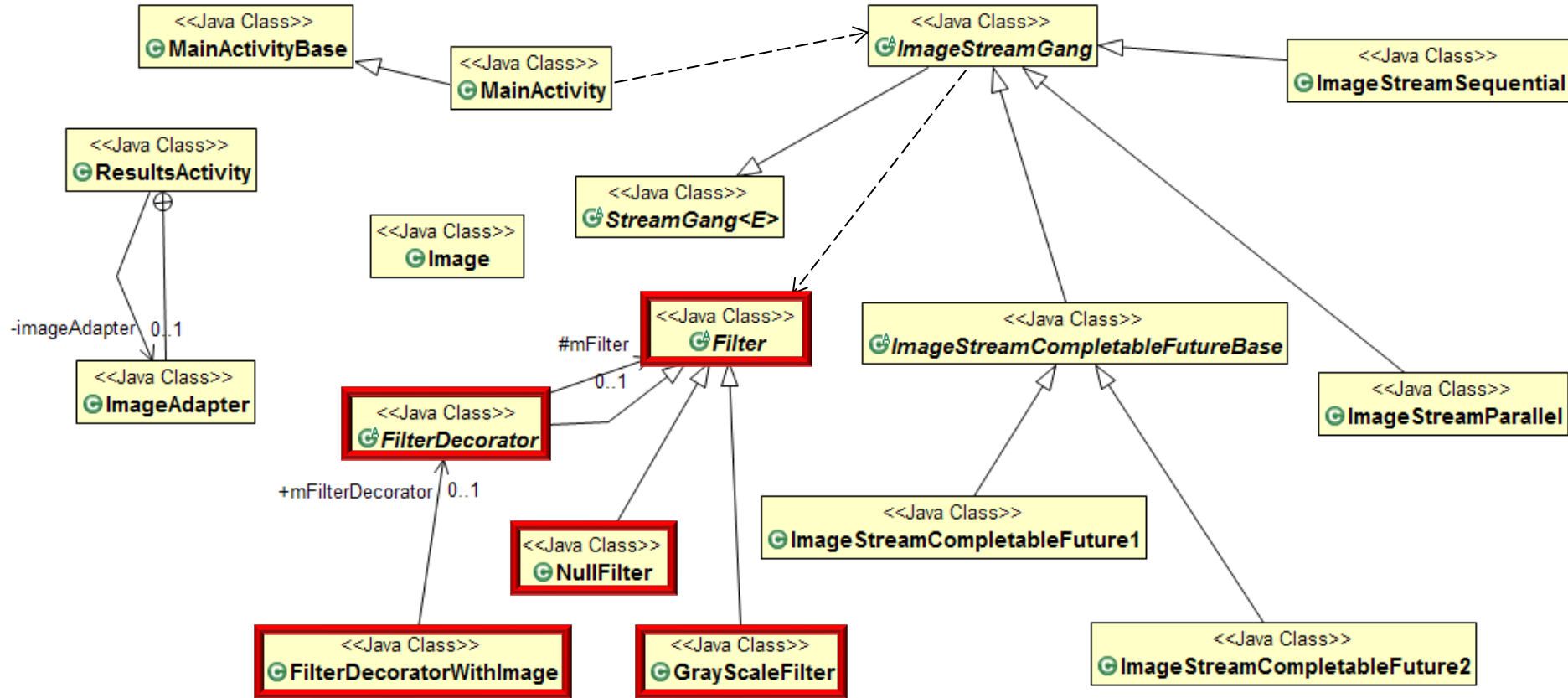
- UML class diagram for the object-oriented ImageStreamGang app design



Stores image meta-data & provides methods for common image-/file-related tasks

# The Structure of the ImageStreamGang App

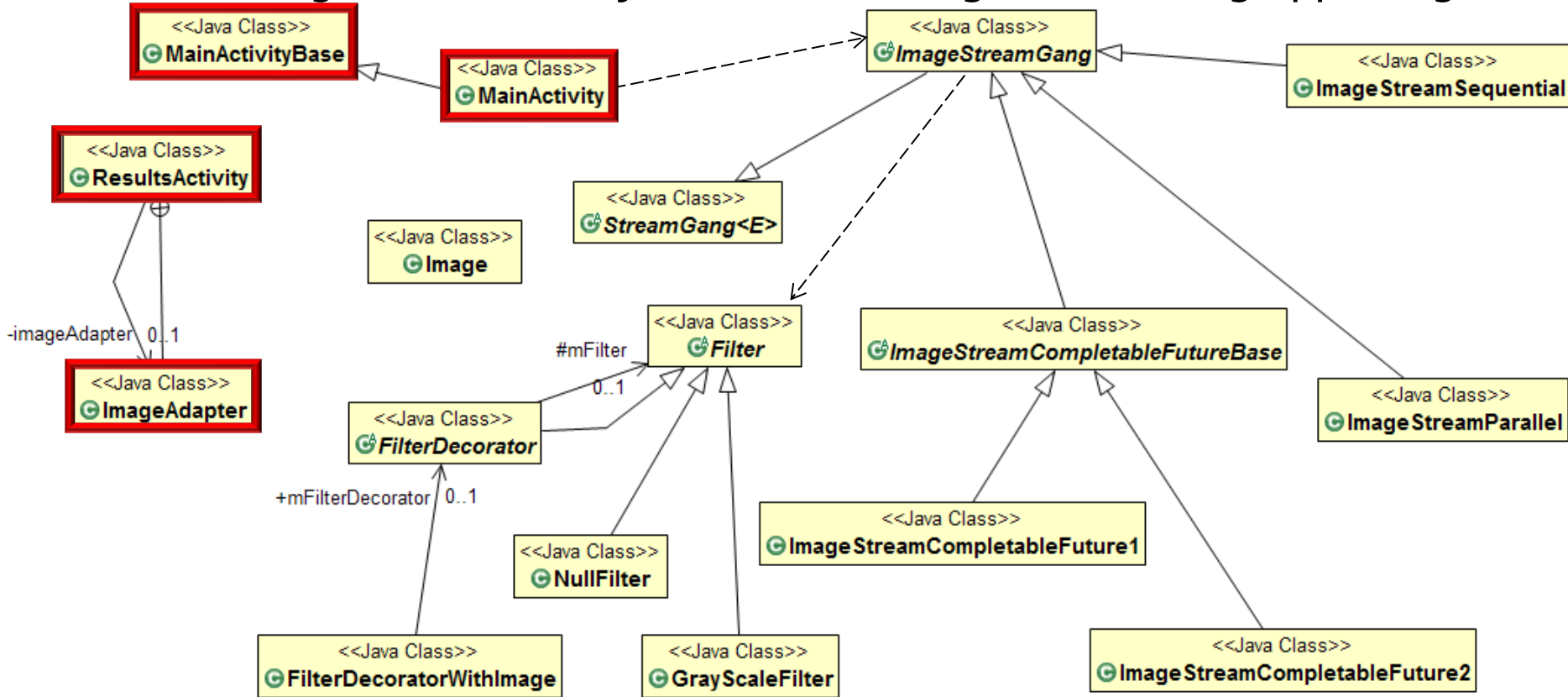
- UML class diagram for the object-oriented ImageStreamGang app design



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

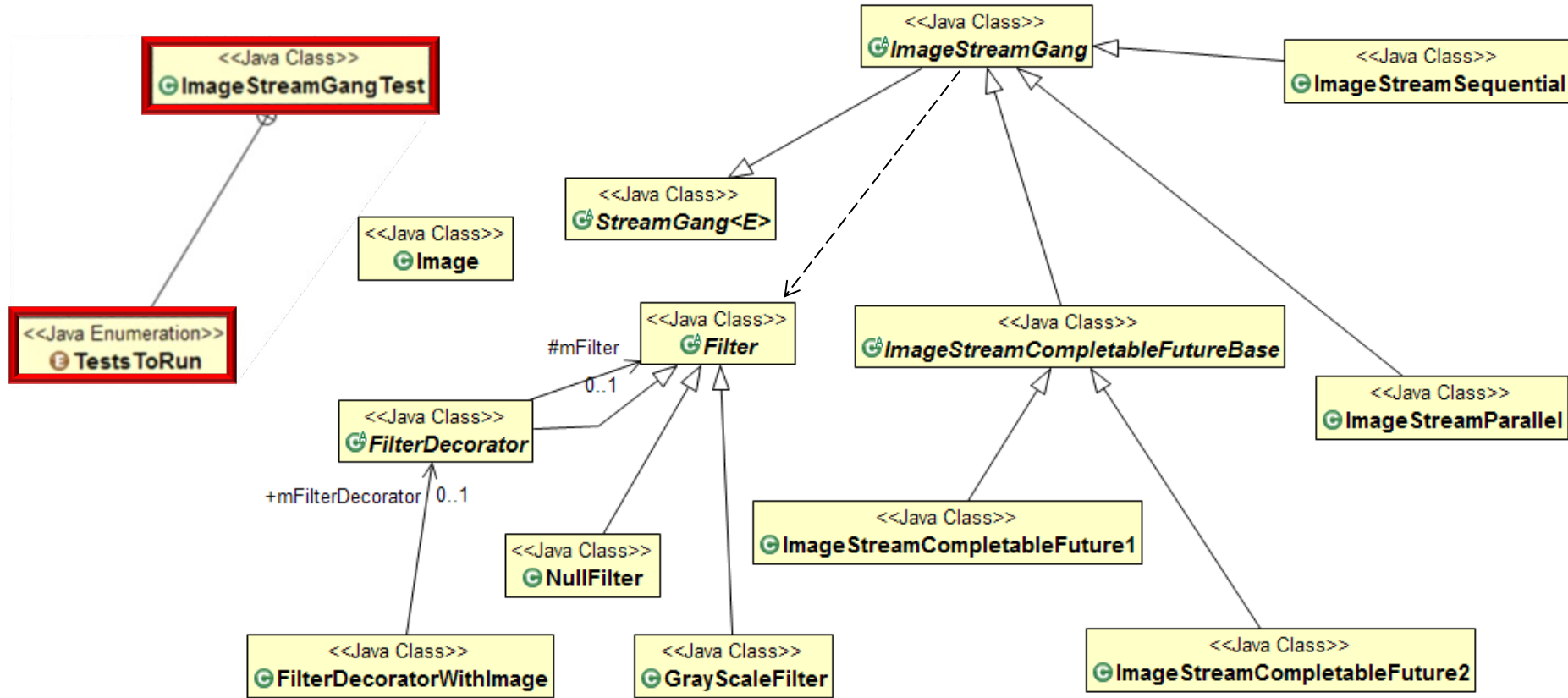


Provides the user interface for an Android app



# 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

# Running the ImageStreamGang App

Starting ImageStreamGangTest

Printing 4 results for input file 1 from fastest to slowest

COMPLETABLE\_FUTURES\_2 executed in 153 msecs

COMPLETABLE\_FUTURES\_1 executed in 251 msecs

PARALLEL\_STREAM executed in 300 msecs

SEQUENTIAL\_STREAM executed in 1026 msecs

Printing 4 results for input file 2 from fastest to slowest

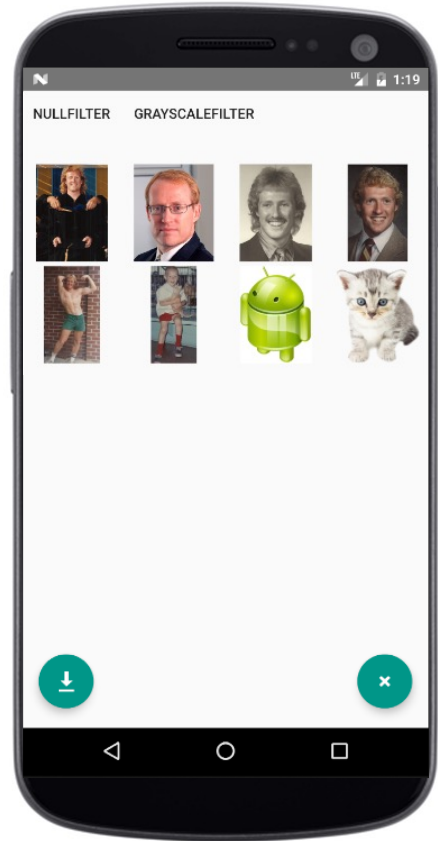
PARALLEL\_STREAM executed in 62 msecs

COMPLETABLE\_FUTURES\_1 executed in 68 msecs

COMPLETABLE\_FUTURES\_2 executed in 70 msecs

SEQUENTIAL\_STREAM executed in 261 msecs

Ending ImageStreamGangTest



Tests conducted on a 2.4 GHz eight-core Lenovo P1 with 128 Gbytes of RAM

---

# End of Understand the Java Parallel ImageStreamGang App Structure