Android Concurrency Frameworks: Structure & Functionality

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

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

• Know the motivations for Android concurrency & concurrency frameworks

• Recognize the two types of Android concurrency frameworks

• Understand the structure & functionality of Android’s concurrency frameworks
Elements of Android Concurrency Frameworks
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces

We focus on classes/interfaces used to write concurrent Android programs
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces

We’ll also outline the implementation of Android’s concurrency frameworks
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces

- **Looper**
  - Run a message loop for a thread
  ```java
  public static void loop() {
    final Looper me = myLooper();
    MessageQueue queue = me.mQueue;
    ...
    for (;;) {
      Message msg = queue.next();
      ...
      msg.target.
      dispatchMessage(msg);
      ...
    } ...
  }
  ```

Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - **Looper**
  - **MessageQueue**
    - Holds the list of messages to be dispatched by a looper

Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - **Looper**
  - **MessageQueue**
    - Holds the list of messages to be dispatched by a looper
    - Messages aren’t added directly to a message queue, but rather via handler objects associated with the looper
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - **MessageQueue**
    - Holds the list of messages to be dispatched by a looper
      - Messages aren’t added directly to a message queue, but rather via handler objects associated with the looper
    - The looper blocks on the message queue until the next message is available
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - **Message**
    - Contains data & type information that can be sent to a handler via a message queue

Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - Message
  - **Handler**
    - Send/process messages & runnables in the message queue associated with a thread’s looper

See developer.android.com/reference/android/os/Handler.html
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - Message
  - Handler
    - Send/process messages & runnables in the message queue associated with a thread’s looper
    - Plays the roles of a proxy for client thread & of a target adapter for dispatching in another thread

See [en.wikipedia.org/wiki/Proxy_pattern](en.wikipedia.org/wiki/Proxy_pattern) & [en.wikipedia.org/wiki/Adapter_pattern](en.wikipedia.org/wiki/Adapter_pattern)
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - Message
  - Handler

- **Runnable**
  - Represents a command that can be executed

See developer.android.com/reference/java/lang/Runnable.html
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - Message
  - Handler

- **Runnable**
  - Represents a command that can be executed
    - This command is often run in a thread different than it was created in

See [www.dre.vanderbilt.edu/~schmidt/PDF/CommandProcessor.pdf](http://www.dre.vanderbilt.edu/~schmidt/PDF/CommandProcessor.pdf)
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - Message
  - Handler
  - Runnable

These classes are used by both the HaMeR & AsyncTask concurrency frameworks

The HaMeR framework exposes some classes to app developers directly, whereas the AsyncTask framework shields app developers from these classes.
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - Message
  - Handler
  - Runnable

Some classes are just used by the AsyncTask concurrency framework
Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - Message
  - Handler
  - Runnable
- **FutureTask**
  - Can be used to
    - Start & cancel a computation that runs asynchronously
    - Query to see if a computation is done
    - Retrieve the result of the computation

Elements of Android Concurrency Frameworks

- Android’s concurrency frameworks are built using reusable classes & interfaces
  - Looper
  - MessageQueue
  - Message
  - Handler
  - Runnable
  - FutureTask
- **Executor framework**
  - Execute submitted runnable tasks either
    - Sequentially in one thread (in the background) or
    - Concurrently in a thread pool

See developer.android.com/reference/java/util/concurrent/Executor.html
These framework elements are used by Android’s application frameworks & packaged/3rd-party applications.
End of Android Concurrency Frameworks: Structure & Functionality
Discussion Questions

1. Which of the following class/interface elements are used by both the HaMeR framework & the AsyncTask framework?
   a. Executor
   b. MessageQueue
   c. Runnable
   d. Message
   e. Handler
   f. FutureTask
   g. Looper