The AsyncTask Framework: Example Application

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

• Recognize the capabilities provided by the Android AsyncTask framework
• Know which methods are provided by AsyncTask class
• Understand what black-box & white-box framework are... & how AsyncTask implements both types of frameworks
• Learn how the AsyncTaskInterrupted program works
Runtime Behavior of the AsyncTaskInterrupted App
• Use AsyncTasks & a ThreadPoolExecutor to compute the greatest common divisor (GCD) of two numbers, which is the largest positive integer that divides two integers without a remainder.
Runtime Behavior of the AsyncTaskInterrupted App

- Use AsyncTasks & a ThreadPoolExecutor to compute the greatest common divisor (GCD) of two numbers, which is the largest positive integer that divides two integers without a remainder.

The user can cancel AsyncTask computations at any time.
Runtime Behavior of the AsyncTaskInterrupted App

• Use AsyncTasks & a ThreadPoolExecutor to compute the greatest common divisor (GCD) of two numbers, which is the largest positive integer that divides two integers without a remainder.

The device’s runtime configuration can also change at any time without affecting running computations.
Implementation of the AsyncTaskInterrupted App
Implementation of the AsyncTaskInterrupted App

- This app showcases the black-box & white-box frameworks in Android’s AsyncTask class

See [github.com/douglascraigschmidt/POSA/tree/master/ex/M5/GCD/AsyncTaskInterrupted](github.com/douglascraigschmidt/POSA/tree/master/ex/M5/GCD/AsyncTaskInterrupted)
Implementations of the AsyncTaskInterrupted App

- This app showcases the black-box & white-box frameworks in Android’s AsyncTask class

Super class automatically logs lifecycle hook method calls to aid debugging
Implementation of the AsyncTaskInterrupted App

- This app showcases the black-box & white-box frameworks in Android’s AsyncTask class.

Start & cancels AsyncTasks that repeatedly compute GCD of two random #'s.
Implementation of the AsyncTaskInterrupted App

- This app showcases the black-box & white-box frameworks in Android’s AsyncTask class.
Implementation of the AsyncTaskInterrupted App

- This app showcases the black-box & white-box frameworks in Android’s AsyncTask class

Extends AsyncTask & in a ThreadPoolExecutor thread repeatedly computing the GCD of two numbers in a manner that can be cancelled at any point.
Implementation of the AsyncTaskInterrupted App

• We’ll now analyze the source code for this app

```java
public class GCDAsyncTask extends AsyncTask<Integer, String, Boolean> {
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        // Passed to doInBackground()
        Integer count;
    }

    @Override
    protected void onProgressUpdate(Integer... values) {
        super.onProgressUpdate(values);
        // Passed to onProgressUpdate()
        String message = String.format("count: %d", count);
    }

    @Override
    protected void onPostExecute(Boolean result) {
        super.onPostExecute(result);
        // Returned from doInBackground()
        // and passed to onPostExecute()
        result.setMessage("Done.");
    }
}
```

```java
public class MainActivity extends LifecycleLoggingActivity {
    String TAG = getClass().getSimpleName();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mAsyncTask = new GCDAsyncTask();
        mAsyncTask.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
    }
}
```

See github.com/douglascraigschmidt/POSA/tree/master/ex/M5/GCD/AsyncTaskInterrupted
End of the AsyncTask Framework: Example Application