Types of Frameworks in AsyncTask (Part 2)

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 Module

• Understand what black-box & white-box framework are... & how AsyncTask implements both types of frameworks

• Learn how the AsyncTaskInterrupted program works

See [github.com/douglascraigschmidt/POSA/tree/master/ex/M4/GCD/AsyncTaskInterrupted](https://github.com/douglascraigschmidt/POSA/tree/master/ex/M4/GCD/AsyncTaskInterrupted)
Runtime Behavior of the AsyncTaskInterrupted App
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.
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/M4/GCD/AsyncTaskInterrupted](github.com/douglascraigschmidt/POSA/tree/master/ex/M4/GCD/AsyncTaskInterrupted)
Implementation of the AsyncTaskInterrupted App

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

Super class that automatically logs lifecycle hook method invocations to aid debugging

```java
public class LifecycleLoggingActivity {
    LifecycleLoggingActivity()
    onCreate(Bundle): void
    onStart(): void
    onResume(): void
    onPause(): void
    onStop(): void
    onDestroy(): void
    onSaveInstanceState(Bundle): void
    onRestoreInstanceState(Bundle): void
}
```

```java
public class MainActivity {
    MainActivity()
    onCreate(Bundle): void
    initializeViews(): void
    setCount(View): void
    startOrStopComputations(View): void
    startComputations(int): void
    interruptComputations(): void
    done(): void
    println(String): void
    onRetainNonConfigurationInstance(): Object
    onDestroy(): void
}
```

```java
public class GCDAsyncTask {
    GCDAsyncTask(MainActivity, int, Random)
    setActivity(MainActivity): void
    computeGCD(int, int): int
    onPreExecute(): void
    doInBackground(Integer[]): Void
    onPostExecute(Void): void
    onCancelled(Void): void
}
```

```java
public class AsyncTaskRelatedState {
    AsyncTaskRelatedState()
    mTaskList 0..*
    mActivity 0..1
    mAsyncTaskRelatedState 0..1
}
```
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 the GCD of two random numbers
Implementation of the AsyncTaskInterrupted App

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

Stores state (including the AsyncTasks & ThreadPoolExecutor) that’s passed between instances of the MainActivity after runtime configuration changes
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.
End of Types of Frameworks in AsyncTask (Part 2)
Discussion Questions

1. Which of the following are characteristics of white-box & black-box frameworks in the AsyncTaskInterrupted app?

   a. *Passing the ThreadPoolExecutor object to executeOnExecutor() is an example of a black-box framework technique*

   b. *Passing the ThreadPoolExecutor object to executeOnExecutor() is an example of a white-box framework technique*

   c. *GCDAsyncTask extending AsyncTask is an example of a white-box framework technique*

   d. *GCDAsyncTask extending AsyncTask is an example of a black-box framework technique*