ToiletPaper #121

Fröhlich zwitschern die Leaks

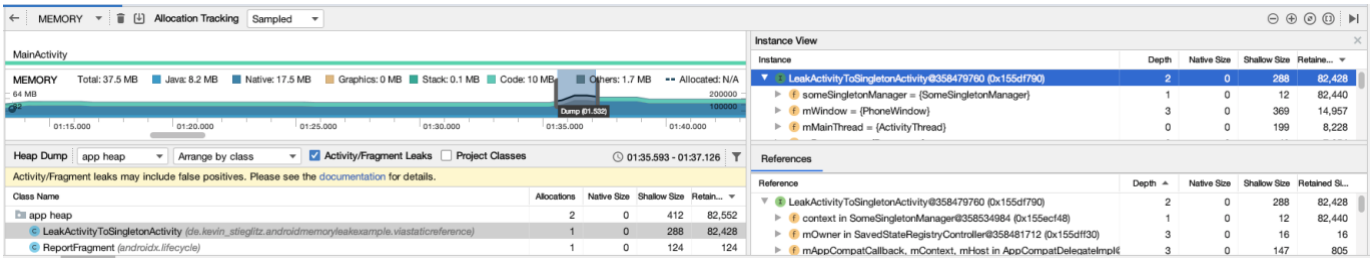
Autor: Kevin Stieglitz / Software Engineer / Business Division Automotive World

✗ Problem

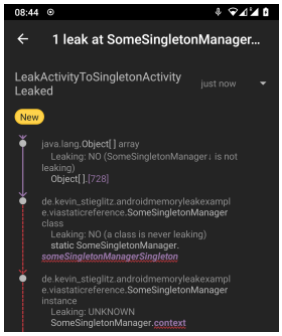
Niemand mag Memory Leaks. Sie sind meist schwer zu reproduzieren, noch schwerer zu identifizieren und machen sich im schlimmsten Fall erst mit einer OutOfMemoryException nach langer Benutzung bemerkbar.

✓ Lösung

Unter Android gibt es glücklicherweise mehrere Möglichkeiten, Leaks zu erkennen. Eine Möglichkeit hierzu bietet Android Studio bereits intern an: Mit Hilfe des Profilers kann der Java Heap angezeigt und anschließend analysiert werden. Seit Android Studio 3.6 kann dieser auch speziell nach nicht mehr existenten, aber immer noch referenzierten Activities/Fragments gefiltert werden (Activity/Fragment Leaks). Dabei werden neben den bestehenden Instanzen auch die Referenzen auf selbige angezeigt, wodurch Rückschlüsse auf Leaks gezogen werden können. Der unten angezeigte Screenshot zeigt eine Beispielanwendung, bei welcher innerhalb eines Singleton die Referenz auf eine Activity gehalten wird, wodurch diese anschließend vom Garbage Collector nicht mehr aufgeräumt werden kann.



Eine weit komfortablere Lösung bietet die Library **LeakCanary** an. Bei dieser wird in Debug-Builds eine WeakReference auf verschiedene Instanzen erstellt (beispielsweise auf Activities in der onActivityDestroyed-Methode). Ein Background-Thread überprüft anschließend, ob die Referenz (nach dem Ausführen des Garbage Collectors) aufgeräumt wurde. Falls nein wird der Heap in einer .hprof-Datei gespeichert. Dieser wird in einem separaten Prozess unter Zuhilfenahme der vorherigen WeakReference analysiert und die Kette an Referenzen berechnet, welche das Objekt davon abhalten, seinen Speicher freizugeben. Mögliche Ursachen für den Memory-Leak werden dabei von LeakCanary durch eine rote Wellenlinie hervorgehoben. Somit ist es auf recht einfache Weise möglich, Leaks bereits frühzeitig zu erkennen und deren Herkunft effizient einzugrenzen.



+ Weiterführende Aspekte

- Android-App mit LeakCanary und einer Vielzahl von gängigen Leaks: https://github.com/kvn-stgl/AndroidMemoryLeakExample
- LeakCanary: https://square.github.io/leakcanary/