Evaluating the Pros & Cons of TimedMemoizer

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

Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

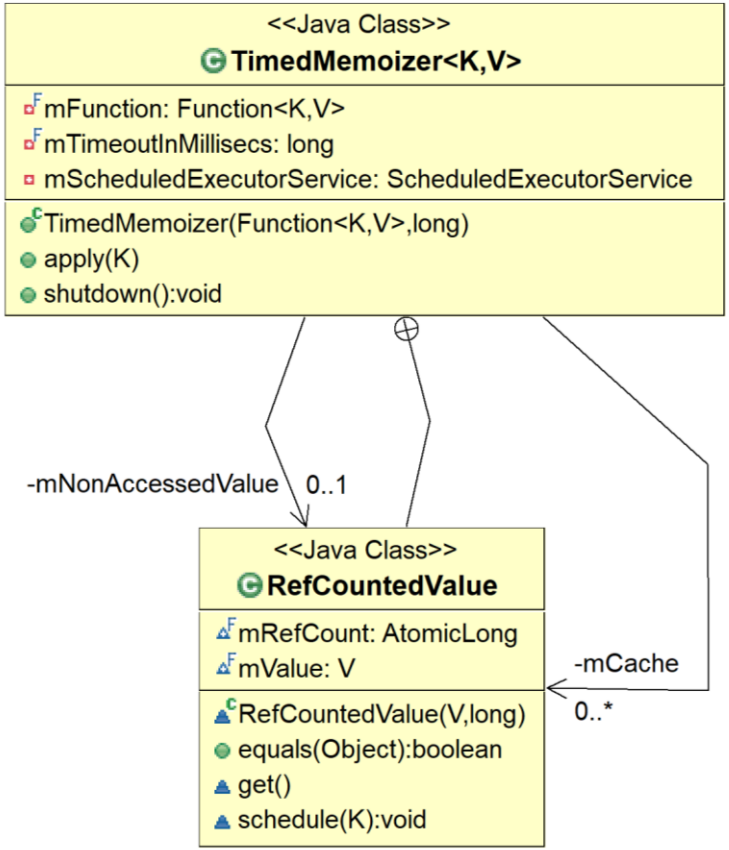
Learning Objectives in this Part of the Lesson

- Learn how to create a TimedMemoizer that applies ScheduledExecutorService to remove stale entries
- Know how to implement the TimeMemoizer class
- Evaluate the pros & cons of the TimedMemoizer class

Evaluating the TimedMemoizer Class

Evaluating the TimedMemoizer Class

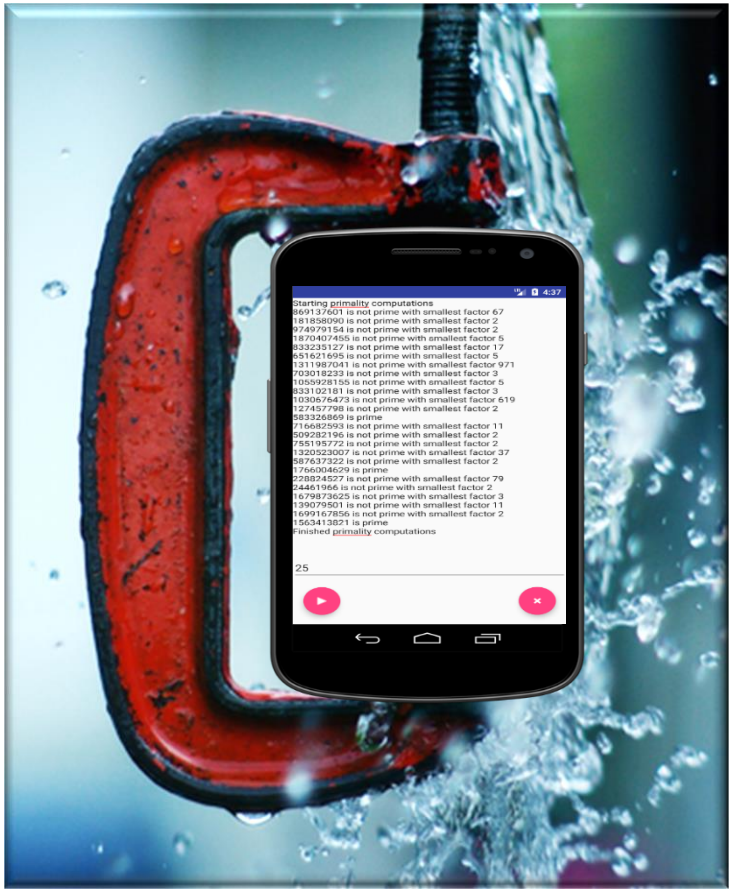
- If a TimedMemoizer is used over time the ScheduledExecutorService will clean it up periodically



This cleanup keeps memory usage from expanding indefinitely

Evaluating the TimedMemoizer Class

- However, there is a limitation



Evaluating the TimedMemoizer Class

- However, there is a limitation
  - TimedMemoizer creates a runnable for each key it passes to the Scheduled ExecutorService, which does not scale well

We'll fix this problem in the upcoming lesson on "*Implementing TimedMemoizerEx*"

End of Evaluating the Pros & Cons of TimedMemoizer