This course has been archived (Saturday, 17 December 2022, 00:00).

Course
CS-E4110
Course materials
Your points
MyCourses
Zulip Chat

This course has already ended.
The latest instance of the course can be found at: Concurrent Programming: 2023

« 3 Atomic Integer -- Atomic operation basics using Atomic Integer. Course materials 5 Feedback »

CS-E4110 / Round 1 - Scala concurrency - Part 1 / 4 Atomic Reference -- Atomic operation basics using Atomic Reference.

Assignment description | My submissions (1/10)

Atomic Reference -- Atomic operation basics using Atomic Reference.

Atomic Reference

One issue that needs to be noted when implementing concurrent programs is memory consistency and errors associated with it. We say memory consistency error occurred when multiple execution units operating on a shared memory have an inconsistent view of a given shared variable. There are a lot of reasons why these inconsistencies happen which include caching and compiler optimizations. The key to avoiding memory consistency problems is to understand and ensure happens-before relationship such that if a happens-before b, the effects of a will be visible to b.

Code

Download the assignment template here

Task

In this exercise, we implement a simple atomic reference; a reference value that may be updated atomically and provides a happens-before relationship when reading and writing a value.

Hint

You may use synchronized keyword to implement happens-before. Visit the basic idea from here. Note also that synchronization is one of the low-level concurrency mechanisms that Scala inherited from Java.

SimpleAtomicReference.scala

Choose File No file chosen

Submit

Earned points
25 / 25

Exercise info

Assignment category
Programming exercises

Your submissions
1 / 10

Deadline
Friday, 12 November 2021, 14:00

Late submission deadline
Friday, 19 November 2021, 14:00 (-30%)

Total number of submitters
56

« 3 Atomic Integer -- Atomic operation basics using Atomic Integer. Course materials 5 Feedback »