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

Assignment description My submissions (3/10)

Futures and Promises-- High-level mechanisms for asynchronous programming.

IMap

In the previous exercises, we have seen low-level abstractions such as threads that support asynchronous computation where executions can occur independently of the main program flow. However, using these facilities directly for composing asynchronous computations is cumbersome and error-prone. In this exercise, we will focus on two abstractions in Scala that are specifically tailored for such tasks called futures and promises. In short, a promise and a future represent two aspects of a single-assignment variable. The promise allows you to assign a value to the future object, whereas the future allows you to read that value.

Code

Download the assignment template here

Task

In this task, we implement a single-assignment map with a twist:

```
class IMap[K, V] {
    def update(k: K, v: V): Unit
    def apply(k: K): Future[V]
    def whenReady[T: scala.reflect.ClassTag](func: V => T , p: Promise[T], f: Future[V])
}
```

Pairs of keys and values can be added to the IMap object, but they can neither be removed or modified. A specific key can be assigned only once, and subsequent calls to update with that same key should result in an exception. Calling apply with a specific key should return a future, which is completed after that key is inserted into the map. As an additional learning twist, we will also implement a whenReady function which completes a promise p using a function func (i.e. p success func(v)) when the future f is completed with a value v (use a callback to check f's completion).

Hint

Use futures and promises. Consult lecture 4. Remember also to consult the Scala documentation, especially, this tutorial. You may also find help from here. If you want a deeper view, see Chapter 4 of the book: Learning concurrent programming in Scala by Aleksandar Prokopec.

IMap.scala

Choose File No file chosen

Submit

Earned points
35 / 35

Exercise info

Assignment category
Programming exercises

Your submissions
3 / 10

Deadline
Wednesday, 8 December 2021, 14:00

Late submission deadline
Wednesday, 15 December 2021, 14:00 (-30%)

Total number of submitters
48