Open Learning Library My Courses All Courses About Contact Give Help vitorpbarbosa7

Course Progress

Course › Readings › Reading 6: Thread Safety › Questions

Previous Next

Questions

Bookmark this page

Serializability

5/5 points (graded)

Suppose two threads are sharing a `MyStringBuffer` representing "z".

For each pair of concurrent calls and their result, does that outcome violate serializability (and therefore demonstrate that `MyStringBuffer` is not threadsafe)?

`clear()` and `insert(0, "a")` → `insert` throws an `IndexOutOfBoundsException`

● Violates serializability

○ Consistent with serializability

✓

Explanation
Calling `insert` with `pos=0` should never violate its precondition, so throwing the exception is not consistent with any sequential ordering.

`clear()` and `insert(1, "a")` → `insert` throws an `IndexOutOfBoundsException`

○ Violates serializability

● Consistent with serializability

✓

Explanation
This is consistent with clear-followed-by-insert. The precondition pos <= length of the current string is violated, since clear happened first.

`first()` and `insert(0, "a")` → `first` returns "a"

○ Violates serializability

● Consistent with serializability

✓

Explanation
This is consistent with insert-followed-by-first.

`first()` and `clear()` → `first` returns "z"

○ Violates serializability

● Consistent with serializability

✓

Explanation
This is consistent with first-followed-by-clear.

`first()` and `clear()` → `first` throws an `IndexOutOfBoundsException`

● Violates serializability

○ Consistent with serializability

✓

Explanation

The exception is never allowed by `first` 's postcondition, so throwing it is not consistent with any sequential ordering.

Submit

Show Answer

Answers are displayed within the problem

< Previous | Next >

Open Learning Library

About

Accessibility

All Courses

Why Support MIT Open Learning?

Help

Connect

Contact

Twitter

Facebook

Privacy Policy Terms of Service