Concurrent Object Oriented Languages

java.util.concurrent.atomic

wiki.eecs.yorku.ca/course/6490A

Presentations of Assignment 2

Schedule the presentations.

java.util.concurrent.atomic

The Java package `java.util.concurrent.atomic` contains classes that support lock-free thread-safe programming on single variables.

AtomicReference<V>

Objects of type `AtomicReference<V>` contain a value of type `V` that may be updated atomically.

The class contains the method

```
public final boolean compareAndSet(V expect,
                                   V update)
```

It atomically sets the value to `update` if the current value of the object == `expect`. It returns true if the update is successful, and false otherwise.

AtomicReference<V>

The class `AtomicReference<V>` contains the method

```
public final V get()
```

It returns the current value of the object.

AtomicReference<V>

Problem

Implement a Stack by means of AtomicReference<V>.

Problem

Implement a Node class.

Stack

```
pop():
success = false;
while not success do
  node = top;
  if (node == null) throw an exception;
  success = CAS(top, node, node.getNext());
return element of node;
```

Stack

```
push(element):
node = node with element;
success = false;
while not success do
  node.next = top;
  succes = CAS(top, node.getNext(), node);
```

AtomicReferenceFieldUpdater<T,V>

The class **AtomicReferenceFieldUpdater<T,V>** contains the static method

```
public static <U,W> AtomicReferenceFieldUpdater<U,W
   newUpdater(Class<U> classType,
              Class<W> attributeType,
              String attributeName)
```

It returns an object that can be used to atomically update the attribute with the given **attributeName**.

AtomicReferenceFieldUpdater<T,V>

The class `AtomicReferenceFieldUpdater<T,V>` contains the method

```
public abstract boolean compareAndSet(T object,
                                      V expect,
                                      V update)
```

It atomically sets the attribute of the given `object` managed by this updater to the given `update` value if the current value === `expect`.

This method is guaranteed to be atomic with respect to other calls to `compareAndSet`, but not necessarily with respect to other changes to the attribute.

AtomicReferenceFieldUpdater<T,V>

Problem

Implement a Stack by means of AtomicReferenceFieldUpdater<T,V>.

AtomicInteger

The class `AtomicInteger` contains methods such as

```
public final int incrementAndGet()
```

and

```
public final int getAndAdd(int delta)
```

AtomicInteger

```
private AtomicInteger counter;

public int getNext()
{
  return counter.getAndIncrement();
}
```

AtomicStampedReference<V>

The class **`AtomicStampedReference<V>`** not only manipulates an object, but also an integer stamp. The class contains the method

```
public boolean compareAndSet(V expectedReference,
                             V newReference,
                             int expectedStamp,
                             int newStamp)
```

AtomicStampedReference<V>

The class `AtomicStampedReference<V>` can be used when you want to manipulate an object atomically, but only when it is in a particular state (the state of the object should be represented as an integer).

No lectures on

- Thursday October 29 and
- Tuesday November 3.