

433 Final Exam, Fall 2005

December 20th, 2005

Answers

1. In his talk "Growing a Language", what did Guy Steele say about adding complex numbers to a programming language such as Java?

Answer: That there are other language features, such as rational numbers, that are equally worthy of inclusion. Rather than including either or both, a language should include neither but provide a way for programmers to write libraries/extensions for complex numbers and rational numbers that work as though they were part of the language.

Thus, by allowing programmers to define their own words, they will be able to grow their language.

2. Say you have an interface A, and classes AImpl and B, where AImpl implements A and B uses A (e.g., B invokes methods on objects implementing the A interface) but not vice-versa (AImpl objects never invoke methods on B objects).

We want to change the system to allow A to be implemented both by AImpl and by remote proxies of AImpl (these might be hand written, as in our project 4, or automatically produced using something like Java RMI).

At a high level, describe the coding and design changes we might have to make to B in order to accommodate the fact that an A reference might now point to either a AImpl or a remote proxy for an AImpl. If you wish, you can also discuss any changes that should be made to the A interface.

Answer: Methods invoked on a remote proxy might fail unexpectedly (because we can't connect to the machine the AImpl lives on) or be very slow.

Thus, any place where Y invokes a method on an X, we need to be prepared for a exception to be thrown or a lengthy delay. This means error handling, and we may want to invoke the method in a way that if one particular remote proxy is responding very slowly, it doesn't block other operations.

It might be a good idea to change the X interface so that all methods are declared as throwing a checked exception. This would ensure that when code invokes a method on an X object, we have written error handling code.

3. What is the most fundamental difference between the SAX and DOM APIs? Under what circumstance would using SAX generally be a much better solution than DOM? Under what circumstance would DOM generally be a much better solution SAX?

Answer: SAX is a streaming, event based API for reading XML, while DOM reads an entire XML document into memory and creates a tree that can be traversed. SAX is good if you need to process large XML documents or if you need high efficiency, and can do your processing in a single pass over the document. DOM is good if you need to do processing more complicated than a single pass over the tree, and your XML document can comfortably fit into memory (the DOM representation will be several times larger than the character representation).

4. `ConcurrentHashMap<K,V>` introduces a new function

```
public V putIfAbsent(K key,V value)
```

Provide a static method for a utility class

```
public static <K,V> V putIfAbsent(Map<K,V> map, K key, V value)
```

that tries to provide the same functionality as best as possible for any Map Don't use any casts. If it does not provide exactly the same functionality as the `putIfAbsent` defined in `ConcurrentHashMap`, explain why.

Answer: Here is an method that works in a single threaded context.

```
public static <K,V> V putIfAbsent(Map<K,V> map, K key, V value) {  
    V v = map.containsKey(key);  
    if (v != null) return v;  
    map.put(key, value);  
    return null;  
}
```

This won't work in a multithreaded context, because the `containsKey` and `put` operations will not be performed atomically. You could try locking the map:

```
public static <K,V> V putIfAbsent(Map<K,V> map, K key, V value) {  
    synchronized(map) {  
        V v = map.containsKey(key);  
        if (v != null) return v;  
        map.put(key, value);  
        return null;  
    }  
}
```

but that won't provide thread safety because not all thread safe Maps block by synchronizing on the map object. In particular, `ConcurrentHashMap` doesn't perform synchronization on the that way, and there is no way to prevent other updates from occurring to a `ConcurrentHashMap`. Thus, any sequences of operations that need to be performed atomically on a `ConcurrentHashMap` need to be supported as a primitive operation.

5. LockSet **answer:**

```
public class LockSet {

    private Lock realLock = new ReentrantLock();
    private Condition waiting = realLock.newCondition();

    public boolean tryLock(Token... locks) {
        realLock.lock();
        try {
            for (Token lck : locks)
                if (lock.locked) return false;
            for (Token lck : locks)
                lock.locked = true;
            return true;
        } finally { realLock.unlock(); }
    }

    public void lock(Token... locks) {
        realLock.lock();
        try {
            while (!tryLock(locks))
                waiting.awaitUninterruptibly();
        } finally { realLock.unlock(); }
    }

    public void lockInterruptibly(Token... locks) throws InterruptedException {
        realLock.lock();
        try {
            while (!tryLock(locks))
                waiting.await();
        } finally { realLock.unlock(); }
    }

    public void unlock(Token... locks) {
        realLock.lock();
        try {
            for (Token lck : locks)
                lck.locked = false;
            waiting.signalAll();
            return;
        } finally { realLock.unlock(); }
    }

    public final class Token {
        boolean locked = false;

        Token() {}

        public Token makeLock() {
            return new Token();
        }
    }
}
```

6. XML: Here is an XML DTD named simple.dtd:

```
<!ELEMENT A (B, C*) >
<!ELEMENT B ((D, B) | C+) >
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (#PCDATA)>
```

- (a) Give a well-formed XML document that validates against simple.dtd that does not contain a D element.

Answer:

```
<A><B><C>Hello</C></B></A>
```

- (b) Give a well-formed XML document that validates against simple.dtd that contains a D element. **Answer:**

```
<A><B><D>Hello</D>
  <B><C>World</C></B>
</B>
</A>
```

- (c) Give an XPath query string that will match all C elements. **Answer:** //C
- (d) Give an XPath query string that will match all C elements contained inside a B element. **Answer:** //B/C
- (e) Give an XPath query string that will match all C elements not contained inside a B element. **Answer:** /A/C