public Object nextElement()
{
    if(count == 0)
        throw new NoSuchElementException();
    else
    {
        count--;
        return elements[count];
    }
}

The inner class Enumerator within Stack uses the fields elements and top from Stack. These refer to the fields within the object that created the Enumerator.

To support these, the translation includes a reference to the enclosing object, called this$0. It is initialized in the constructor to the object responsible for the creation of the Enumerator. All references to top and elements come from this reference.

The compilation of Enumerator produces these definitions:

```
.class Stack$Enumerator
   .implements java/util/Enumeration

   .field this$0 LStack;            ; The enclosing object
   .field count I                   ; The current count

   .method <init>(LStack;)V
       aload_0                      ; Call super constructor
       invokespecial java/lang/Object/<init>()V
       aload_0                      ; Store the enclosing object
       aload_1                      ; in this$0
       putfield data/structure/Stack$Enumerator/this$0 LStack;

       aload_0                      ; This is the body of the
       aload_1                      ; constructor:
       getfield Stack/top I         ; count = top;
       putfield count I

       return
   .end method
```