```
public interface Enumeration<E>
{
        public bool hasMoreElements(); // Tells if there are any more
elements in the collection
        public E nextElement(); // Returns the next element in the
collection/enumeration
}


public interface Iterator<E>
{
        public bool hasNext(); // Tells if there are any more elements
in the collection
        public E next(); // Returns the next element in the
collection/iteration.
        public void remove(); // Removes from the underlying
collection the last element returned by this iterator
}


public class EnumerationIterator implements Iterator<Object> {
        Enumeration<?> enumeration;

        public EnumerationIterator(Enumeration<?> enumeration) {
                this.enumeration = enumeration;
        }

        public boolean hasNext() {
                return enumeration.hasMoreElements();
        }

        public Object next() {
                return enumeration.nextElement();
        }

        public void remove() {
                throw new UnsupportedOperationException();
        }
}



public class EnumerationIteratorTestDrive {
        public static void main (String args[]) {
                Vector<String> v = new Vector<String>
(Arrays.asList(args));
                // Pass old style Enumeration to the adapter
                Iterator<?> iterator = new
EnumerationIterator(v.elements());
                // Now we can use the new style  Iterator methods
                while (iterator.hasNext()) {
                        System.out.println(iterator.next());
                }
        }
}
```