Filtered List

We want to make a `FilteredList` class that selects only certain elements of a `List` during iteration. To do so, we’re going to use the `Predicate` interface defined below. Note that it has a method, `test` that takes in an argument and returns `true` if we want to keep this argument or `false` otherwise.

```java
public interface Predicate<T> {
    boolean test (T x);
}
```

For example, if `L` is any kind of object that implements `List<String>` (that is, the standard `java.util.List`), then writing

```java
FilteredList<String> FL = new FilteredList<String> (L, filter);
```

gives an `iterable` containing all items, `x`, in `L` for which `filter.test(x)` is `true`. Here, `filter` is of type `Predicate`. Fill in the `FilteredList` class below.

```java
import java.util.Iterator;
import java.util.NoSuchElementException;
public class FilteredList<T> {
    public FilteredList (List<T> L, Predicate<T> filter) {
    }
    @Override
    public Iterator<T> iterator() {
    }
}
```
2 Iterator of Iterators

Implement an IteratorOfIterators which will accept as an argument a List of Iterator objects containing Integers. The first call to next() should return the first item from the first iterator in the list. The second call to next() should return the first item from the second iterator in the list. If the list contained n iterators, the n+1th time that we call next(), we would return the second item of the first iterator in the list.

For example, if we had 3 Iterators A, B, and C such that A contained the values [1, 2, 3], B contained the values [4, 5, 6], and C contained the values [7, 8, 9], calls to next() for our IteratorOfIterators would return [1, 4, 7, 2, 5, 8, 3, 6, 9]

Feel free to modify the input a as needed.

```java
import java.util.*;

public class IteratorOfIterators {

    public IteratorOfIterators(List<Iterator<Integer>> a) {

    }

    @Override
    public boolean hasNext() {
    }

    @Override
    public Integer next() {
    }
}
```
3

Every $\kappa$th Element (Fall 2014 MT1 Q5)

Fill in the next() method in the following class. Do not modify anything outside of next.

```java
import java.util.Iterator;
import java.util.NoSuchElementException;
/** Iterates over every Kth element of the IntList given to the constructor.
 * For example, if L is an IntList containing elements
 * [0, 1, 2, 3, 4, 5, 6, 7] with K = 2, then
 * for (Iterator<Integer> p = new KthIntList(L, 2); p.hasNext(); ) {
 *     System.out.println(p.next());
 * }
 * would print get 0, 2, 4, 6. */
public class KthIntList implements Iterator<Integer> {
    public int k;
    private IntList curList;
    private boolean hasNext;

    public KthIntList(IntList I, int k) {
        this.k = k;
        this.curList = I;
        this.hasNext = true;
    }

    /** Returns true iff there is a next Kth element. Do not modify. */
    public boolean hasNext() {
        return this.hasNext;
    }

    /** Returns the next Kth element of the IntList given in the constructor.
     * Returns the 0th element first. Throws a NoSuchElementException if
     * there are no Integers available to return. */
    public Integer next() {
        // Your implementation goes here
    }
}
```