

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+1`th time that we call `next()`, we would return the second item of the first iterator in the list.

Note that if an iterator is empty in this process, we continue to the next iterator. Then, once all the iterators are empty, `hasNext` should return **false**. For example, if we had 3 `Iterators` A, B, and C such that A contained the values [1, 3, 4, 5], B was empty, and C contained the values [2], calls to `next()` for our `IteratorOfIterators` would return [1, 2, 3, 4, 5].

```
1 import java.util.*;
2 public class IteratorOfIterators _____ {
3
4
5     public IteratorOfIterators(List<Iterator<Integer>> a) {
6
7
8
9
10
11
12
13     }
14
15     @Override
16     public boolean hasNext() {
17
18
19
20
21     }
22
23
24
25     @Override
26     public Integer next() {
27
28
29
30
31     }
32 }
```