What Is an Iterator?

- An object that traverses a collection of data
- Similar to bookmark
- During iteration, each data item is considered once
  - Possible to modify item as accessed
- Should be implemented as a distinct class that interacts with the collection

Why Iterators?

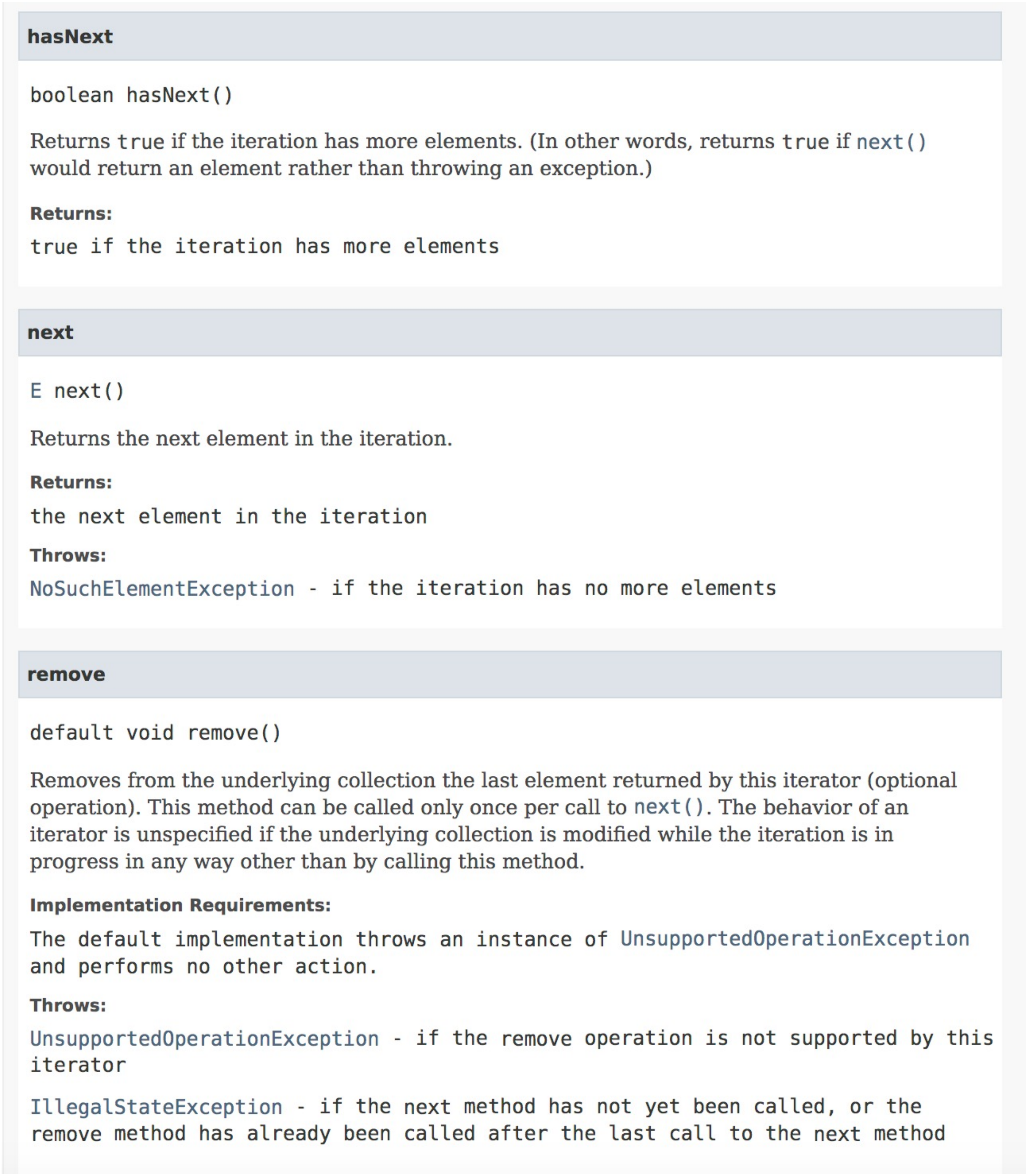
```
ArrayList<Integer> values = new ArrayList<Integer>();
    …
    ///add 6 numbers to values
    …

    // The goal is to remove all even numbers
    for(int i = 0; i < 6 ; i++){
         if (values.get(i) % 2 == 0)
              values.remove(i);
          }

    // Same here. But which one do you think will work?
    for(Iterator<Integer> it = values.iterator(); it.hasNext();)
         if(it.next() % 2 == 0)
               it.remove();


    System.out.println("Result:” + values.toString());
```

$$\begin{array}{cccccc} 1 & \cancel{4} & 18 & 2 & 3 & 7 \\ 0 & \cancel{1} & 2 & 3 & 4 & 5 \\ \uparrow & \uparrow & & & & \end{array}$$

$$\begin{array}{c} i \\ \hline \cancel{0} \\ \cancel{1} \\ 2 \\ 3 \\ 4 \\ \cancel{5} \end{array}$$

$$\begin{array}{ccccc} 1 & 18 & \cancel{2} & 3 & 7 \\ 0 & 1 & \cancel{2} & 3 & 4 \\ & & \uparrow & & \end{array}$$

$$\begin{array}{ccccc} 1 & \bigcirc\!\!\!\!\!\!\begin{smallmatrix}18\\1\end{smallmatrix} & 3 & 7 & \\ 0 & & 2 & 3 & 4 \\ & & & \uparrow & ?. \end{array}$$



hasNext

```
boolean hasNext()
```

Returns `true` if the iteration has more elements. (In other words, returns `true` if `next()` would return an element rather than throwing an exception.)

Returns:

`true` if the iteration has more elements

next

```
E next()
```

Returns the next element in the iteration.

Returns:

the next element in the iteration

Throws:

`NoSuchElementException` - if the iteration has no more elements

remove

```
default void remove()
```

Removes from the underlying collection the last element returned by this iterator (optional operation). This method can be called only once per call to `next()`. The behavior of an iterator is unspecified if the underlying collection is modified while the iteration is in progress in any way other than by calling this method.

Implementation Requirements:

The default implementation throws an instance of `UnsupportedOperationException` and performs no other action.

Throws:

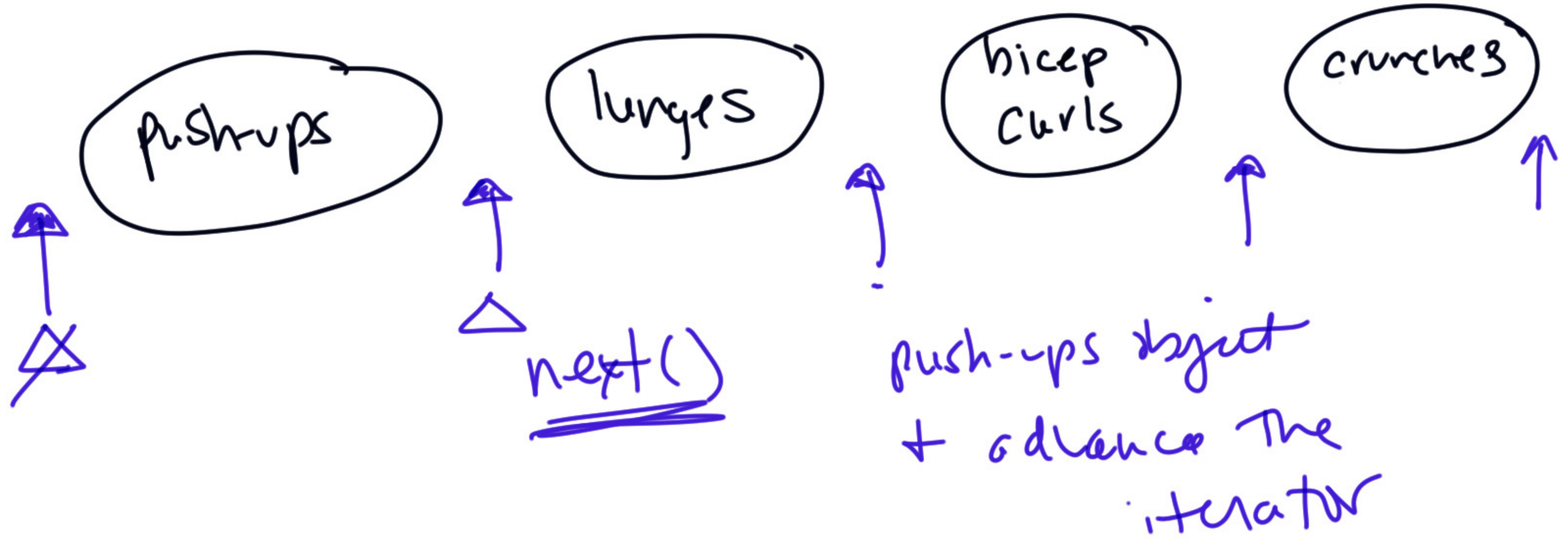
`UnsupportedOperationException` - if the `remove` operation is not supported by this iterator

`IllegalStateException` - if the `next` method has not yet been called, or the `remove` method has already been called after the last call to the `next` method boolean hasNext()

E next()

void remove() ← optional

☆ Think of iterator like a bookmark or cursor, it rests between items



Multiple Iterators