CSM 61B
Spring 2019

Iterators & Exceptions

Mentoring 4: February 18, 2019

1 Iterator Interface

In Java, an **iterator** is an object which allows us to traverse a data structure in linear fashion. Every iterator has two methods: `hasNext` and `next`.

```
interface IntIterator {
    boolean hasNext();
    int next();
}
```

1.1 Consider the following code that demonstrates the `IntArrayIterator`.

```
int[] arr = {1, 2, 3, 4, 5, 6};
IntIterator iter = new IntArrayIterator(arr);
if (iter.hasNext()) {
    System.out.println(iter.next());     // 1
}
if (iter.hasNext()) {
    System.out.println(iter.next() + 3); // 5
}
while (iter.hasNext()) {
    System.out.println(iter.next());     // 3 4 5 6
}
```

Define an `IntArrayIterator` class that works as described above.

```
public class IntArrayIterator __________________ IntIterator {



    public IntArrayIterator(int[] arr) {



    }

    public boolean hasNext() {






    }

    public int next() {






    }
}
```

1.2 Define an `IntListIterator` class that adheres to the `IntIterator` interface.

1.3 Define a method, `printAll`, that prints every element in an `IntIterator` **regardless of how the iterator is implemented.**

2 VoteIterator

2.1 Define `VoteIterator`, an `IntIterator` that takes in an **`int`**`[] array` of vote counts and iterates over the votes. The input array contains the number of votes each candidate received.

```
int[] array = { 0, 2, 1, 0, 1, 0 };
```

Each candidate, represented by their index, `i`, should be returned from each call to `next()` `array[i]` times in total. Given the input above, calls to `next()` would eventually return `1` *twice*, `2` *once*, and `4` *once*.

3 Catch Blocks, Exceptions, and Try-s, Oh My

3.1

```
try {
    doSomething();
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("caught array index exception");
} catch (Exception e) {
    System.out.println("caught an exception");
    throw e;
} catch (NullPointerException e) {
    System.out.println("caught null pointer exception");
} finally {
    System.out.println("in finally block");
}
```

(a) What will print if `doSomething()` throws a `NullPointerException`?

(b) What if `doSomething()` throws an `ArrayIndexOutOfBoundsException`?

(c) What if `doSomething()` doesn't error?