CSE12 - Lecture 22

Friday, May 24, 2024 10:00 AM

No class next Monday (5/27) - holiday
PA8 due Wednesday @ 8am
PA6 Late/Resubmit due Wednesday @ 8am
Exam 3 (6/5)

- https://ucsd-cse12-sp24.github.io/lectures/exam3.html

Lecture 22

Iterators

What is an iterator used for in Java?
Visit, in some order, all elements of a collection
↳ use in a for-each loop for(int i : ars) { } (Iterable)

What is the interface needed for creating an iterator?
Iterable<E> → Iterable<Integer>

What method(s) do we need to implement for that interface?
Iterator<E> iterator() { }
Iterator<Integer> iterator() { }

What class do we need to create to hold the iterators state?
Iterator<E>

Where should that class be created?
private inner class → usually an inner class of our data structure

What interface does it need to implement?
Iterator<E>

What method(s) do we need to implement for that interface?
E next() → Integer next()
boolean hasNext()

What is the process to iterate over an object? (next method)
1. save the current value into a temp variable
2. move to the next item (update state)
3. return the value (temp)

```
class MyClass<E> implements Iterable<E> {
    class MyIterator<E> implements Iterator<E> {
        // state → fields
        public MyIterator(___) {
            // save initial state
        }
        public E next() {
            return null;
        }
        public boolean hasNext() {
            return false;
        }
    }
}
```

```
MyClass<Integer> myC = new ___
for (Integer i : myc) {
    System.out.println(i);
}
```

```
Iterator<Integer> itr = myc.iterator();
while (itr.hasNext()) {
    Integer i = itr.next();
    System.out.println(i);
}
```

Name: _____ PID: _____ Code: 8831

```
// fields of the outer class (usable by inner class)
public Iterator<E> iterator() {
    return new MyIterator(___);
}
}
```

How could we make our linked list work in an enhanced for loop? What changes would we need to malke to the LList class?

```
LList<Integer> list = new LList<Integer>();

//code to add data to list

for (Integer i: list) {
  System.out.println(i);
}
```

```
public class LList<E> {
  Node front;
  int size;

  LList() { //... }
  public void prepend(E value) { //... }
  public E get(int index) { //... }
  public int size() { //... }
```

implements Iterable<E>

boolean changed = false;

changed = true;

```
class Node<E> {
  E value;
  Node<E> next;
  public Node(E value, Node<E> next) {
    this.value = value;
    this.next = next;
  }
}
```

class LLIterator<E> implements Iterator<E> {
// state
Node<E> current;
public MyIterator(__) {
current = front.node;
changed = false;
public boolean hasNext() {
return current != null;
public next() {

}
}
public Iterator<E> iterator() {
return new LLIterator<>();
}

if (changed)
// throw new exception

① E temp = current.value;
② current = current.next;
③ return temp;