CS 367 - Introduction to Data Structures
Tuesday, September 13, 2016

We assume that you are proficient at object-oriented programming in Java.

Course Website: https://canvas.wisc.edu/ (Log in using NetID)
If Canvas is down, we may post updates on: http://pages.cs.wisc.edu/~cs367-1/

See modules links for online readings and lecture outlines (no textbook)
Waitlisted? Sign name, NetID, and UW ID# on sheet at front of class.
WACM Basic Linux Workshop Wed 5:30-7:00pm, Thurs 5:30-7:00pm

Homework h0 and h1, due 10 pm Friday, Sept 16th
Program p1 assigned _________________ (See Assignments)

Assignment questions?
Post it on Piazza. Invitations to join were sent yesterday to your UW email.
Consult with a TA in 1366 CS. See course page for scheduled hours.

Last Time
  Implementing the Bag ADT
    • Java's Object class: casting when using Object
    • Java Generics: using Java generics for generality
  List ADT
    • designing the ListADT
    • coding the ListADT as a Java interface

Today
  Lists
    • using lists via the ListADT
    • implementing the ListADT using an array (SimpleArrayList)
  Java API Lists
  Iterators: Concept

Next Time
  Read: finish Lists
  Iterators
    • iterators and the Java API
    • using iterators
    • options for implementing iterators
    • making a class iterable
Recall the List ADT

Concept

A List is a general, position-oriented container that stores a contiguous collection of items where duplicates are allowed. It maintains relative ordering and uses zero-based indexing.

List Operations

```java
void add(E item);
void add(int pos, E item);
E get(int pos);
E remove(int pos);
boolean contains(E item);
int size();
boolean isEmpty();
```

Issues

Null item – detect then signal with `IllegalArgumentException`

Bad position – detect then signal with `IndexOutOfBoundsException`

Empty list – handle as a bad position
> Assume `myList` is a `ListADT`. What does the following code fragment do in general?

```java
for (int i = 0; i < myList.size(); i++) {
    myList.remove(i);
}
```
→ Assume `myList` is a ListADT. **Write a code fragment** to reverse the contents of `myList` without using any additional ListADTs or other data structures (e.g., array).
Implementation - ListADT using a Generic Array

public class SimpleArrayList<E> implements ListADT<E> {
    private E[] items; //the items in the List
    private int numItems; //the # of items in the List

    public SimpleArrayList() {

    }

    //*** required ListADT methods ***
    public void add(E item) { ... }
    public void add(int pos, E item) { ... }
    public E remove(int pos) { ... }
    public E get (int pos) { ... }
    public boolean contains (E item) { ... }
    public int size() { ... }
    public boolean isEmpty() { ... }

    //*** additional optional array list methods ***
    }

Implementing contains

→ **Complete the method below** so that it returns true iff the given item is in the list.

```java
public boolean contains(E item) {
```
Implementing add at end

What problem might occur with the following implementation:

```java
class Listable {
  private E[] items;
  private int numItems;

  public void add(E item) {
    items[numItems] = item;
    numItems++;
  }
}
```
Java API Lists
Design - Iterators

What are they?

Concept

Operations