

CS367 Lecture 2

Tuesday 17 June 2014

Reminders:

- Register on Piazza
- HW0 due tomorrow at 10 AM
- Email ansari@cs.wisc.edu about VISA, any conflicts, etc.

Last class:

- Course Handout and Intro, Logistics
- Abstraction, Interfaces, and ADTs
- Sack ADT, examples, and array implementation
- What makes software good?

Today:

- Generics
- Lists

Generics

What are they?

- allowing “a type or method to operate on objects of various types while providing compile-time type safety.”

Generic Sack ADT implementation

Benefits

Write code to make a generic Sack that stores **Strings** and another storing **Dies**

List ADT

Conceptual Picture

Operations

```
public interface ListADT<E> {  
    void add(E item);  
    void add(int pos, E item);  
    boolean contains(E item);  
    int size();  
    boolean isEmpty();  
    E get(int pos);  
    E remove(int pos);  
}
```

Error-handling issues

Using the List ADT: Example 1

What does this code fragment do? Suppose L is a list.

```
for (int i = 0; i < L.size(); i++) {  
    L.remove(i);  
}
```

Implementing List ADT: Examples

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

```
public boolean contains(E item)
```

What problem might occur with the following implementation:

```
public void add(E item) {  
    items[numItems] = item;  
    numItems++;  
}
```

Using the List ADT: Example 2

Assume `L` is a List. Write a code fragment to reverse `L` without using an additional List or any other data structure (not even an array).

Implementing List ADT using an array

Lists in Java:

Interface: <http://docs.oracle.com/javase/7/docs/api/java/util/List.html>

Implementation using arrays (ArrayList):

<http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html>