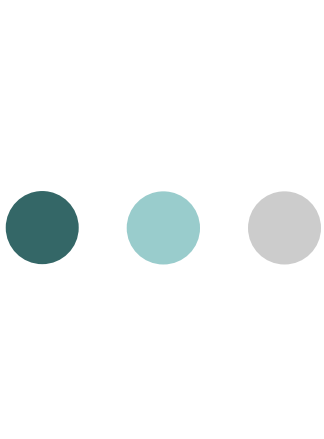

Java API

Lecture 4

Dustin & Mike

Friday, Jan 6, 2006

Overview

- Generics
- Collection and Map
- Collections, Arrays
- Sample Questions

Generics

- Before Java 1.5.0, the world was dark and murky:
- ArrayList a = new ArrayList();
- a.add(new Integer(3));
- int b = ((Integer)a.get(5)).intValue();

- People got occupational diseases from typing toooooooo much.

Generics

- Java 1.5 added generics (similar to templates)
- Combined with auto-de/boxing gives:
- ArrayList<Integer> a = new …
- a.add(3);
- int b = a.get(5);

Generics

- Things to remember:
- Java generic data structures takes objects, not primitive data types
- i.e. Integer instead of int
- Integer, Double, etc. are IMMUTABLE!
- (but they are comparable, hashable, ……all the goodies)
- Use .equals() instead of ==

Collection and Map

- Data structures we'll probably use:
  - ArrayList<E>
  - LinkedList<E> (also for queue, stack)
  - TreeSet<E> (also for priority queue)
  - HashSet<E>
  - TreeMap<K,V>
  - HashMap<K,V>
- First four implements Collection, last two implements Map

Collection and Map

- Common procedures for adding:
  - add(E elem)
  - add(int index, E elem)
  - addAll(Collection<E> c)
  - addFirst(E elem), addLast(E elem)
  - put(K key, V value)
  - putAll(Map<K, V> map)

Collection and Map

- Common mutators:
  - set(int index, E elem)
  - remove(int index)
  - removeFirst(), removeLast()
  - remove(K key)

Collection and Map

- Common queries
  - int size();
  - bool contains(); (containsKey, containsValue)
  - iterator();
  - get(int index)
  - getFirst(), getLast() (linked list)
  - first(), last() (set)

Collection and Map

- To query Maps in Java, it is often useful to turn it into a set:
  - entrySet();
  - keySet();
  - values(); (just a collection)
- For all the Collection data structure, the toArray(); method is also useful

Collections and Arrays

- Some tools for data structures and arrays.
- Tools for data structures are in Collections
- tools for arrays are in Arrays

Collections and Arrays

- Common usage:
  - sort(…);
  - binarySearch(…);
  - fill(…);
  - max(…), min(…);
  - shuffle(…), for randomized algorithms.

Java API

- Lots of things are not mentioned yet:
  - Custom comparators (set, map, sort)
  - tail and head sets
- But no worries. One reference site has it all!
- http://java.sun.com/j2se/1.5.0/docs/api/index.html

Time for Problems!

- Halloween Again!

Sample Problem 1

- FunFun elementary is having a concert. Students need to be sorted by height, with taller people on the left.
- When there is a tie, we sort by name. E.g. Bobby stands to the left of Cindy, even though they are the same height
- Input is a list of names and heights

Sample Problem 2

- Yahoo Text Twist
- Given 6 scrambled letters, make up the correct word with the 6 letters in some order
- Suppose you have a dictionary API, with the method
  - boolean isGoodWord(String s);
- Let’s program the computer so we always win!

Sample Problem 3

- What if it's not 6 letters, but 10?
- Instead, let's say we've lost enough times that we have lots of answers written down, and suppose Yahoo really only have a few problem sets that keep repeating
- e.g. abcdeabcde and edcbaedcba is really the same problem

Sample Problem 3

- After getting a bunch of problem and their solutions, we're showed a bunch of problems without solutions
- If its something we've seen, output the answer. Otherwise, we're stumped.