EECS1012
MOBILE COMPUTING

COLLECTIONS

(SLIDES ADAPTED FROM PROF.H. ROUMANI)

PROF. Y. LESPÉRANCE
Dept. of Electrical Engineering & Computer Science

1

ABOUT COLLECTIONS

- Problem: naming a bunch of things
  Cannot use variables … will run out of names!
- Solutions
  Traditional approach: name + index = array
  Modern approach: object with API = list, set, map
- Comparison
  Arrays have no API and suffer from fixed allocation
  The modern collection framework has a rich API
- But we occasionally use arrays
  For compatibility with low-level API (e.g. split and args)

2

ARRAYS (SEE SEC. L.2.1.E)

- Represent a collection of entities of the same type
- Declaration: *type*[] *name*; e.g. int[] bag;
- Instantiation: *new type*[*size*], e.g.

  bag = new int[100];
- Refer to elements by name[index] , e.g.
  bag[0] = 123; bag[1] = bag[0] + 5;

3

ARRAYS (SEE SEC. L.2.1.E)

- *name.length* represents the array's length
- Indices go from 0 to length – 1
- Multidimensional arrays can also be used

4

Example 1

If we pick an integer in [1,1M] randomly, how likely is it to get one whose digit sum is divisible by 7?

Compute the probability by sampling 10% of those integers and store the sample in a collection.

1. Use Arrays
   See SumDiv7_array.java
2. Use Collections
   See SumDiv7_coll.java

5

Java Collection Framework

- List vs Set vs Map
  List: may contain duplicates and elements are ordered. Set: no duplicates and no order. Map: key-value pairs, key unique.
- The Interfaces (aka Abstract Data Types)
  List<E>, Set<E>, and Map<K,V> (use generics)
- The Classes (aka Implementations)
  List: ArrayList and LinkedyList; Set: HashSet and TreeSet
  Map: HashMap and TreeMap
- Common APIs
  size(), clear(), iterator(), toString()
  Methods to insert, delete, and search → CRUD

6

The Collections API

Basic
size()
clear()
iterator()

List/Set
add(E)
remove(E)
contains(E)

List Only
add(int, E)
remove(int)
get(int)

Map
put(K,V), get(K), keySet()
containsKey(K)
containsValue(V)

Other API
The enhanced for loop
Collections.sort(List)
Arrays… *see API*

7