CS102

Introduction to
data structures, algorithms,
and object-oriented
programming

April 17th, 2017

1

Preview of Lab 9

- Using 2-D arrays to represent matrixes. Writing methods to do matrix operations.
- The operations will be explained...there is no need for you to know linear algebra.

Parameterized types

- We've already seen the ArrayList and Stack types, in which the object to be held in the collection are given in "diamond braces" when the type is declared.
- One of the reasons wrapper classes were created is so primitive types could be stored in containers by using their object-wrapper types.
- An ArrayList<JComponent> can hold objects of type JButton, JPanel, JTextField or any other subclass of JComponent.
- If you look up a parameterized type in the java api, they appear like ArrayList<T>. When T is specified, the compiler ensures that only objects of type T are added.

Methods of the ArrayList class

An ArrayList is also known as a "dynamic array" because it allows constant time random access while at the same time it requires no initial size when instantiated.

size(), add(obj), get(N), set(N, obj), remove(N), remove(obj), indexOf(obj), contains(obj).

You can also use the ArrayList as a non-parameterized type, but that is assumed to be ArrayList<Object>.

Java Collections Framework

An ArrayList is also known as a "dynamic array" because it allows constant time random access while at the same time it requires no initial size when instantiated.

Methods:

size(), add(obj), get(N), set(N, obj), remove(N), remove(obj), indexOf(obj), contains(obj).

Auto-boxing and unboxing

Using a parameterized type such as Integer or Double, you can insert values of type int or double. This is known as autoboxing, because the int is automatically converted to an Integer (same with double to Double and char to Character).

When removing or getting a wrapper type out of a parameterized type, the actual primitive type is returned, a process known as "auto-unboxing".

Vectors

Also parameterized type like ArrayList, but methods are not all the same as ArrayList. This type was around before any of the other parameterized types.

When removing or getting a wrapper type out of a parameterized type, the actual primitive type is returned, a process known as "auto-unboxing".

Methods:
elementAt(N), setElementAt(obj,N), addElement(obj), setSize()

Searching and Sorting

Simplest form of searching algorithm?

Binary search for an element I works on an already sorted list, looking first at the middle element of I and then concentrates on the first part and the last part of I recursively.

Sorting Algorithms

Insertion sort: Maintains the invariant that on iteration i, the i smallest elements are in positions 0...i, they are the elements that were originally in position 0...i, and they are in ascending sorted order.

Keeps first i positions of array sorted and inserts element i+1 into its sorted place in that order.

This is an in-place algorithm because it uses only a constant time extra space besides that needed to store the array.

Sorting Algorithms

Selection sort: Starts at the first element of the array, moves toward the end of the array, and move the biggest element to the end of the array.

Maintains the invariant that, in iteration i, the i largest elements are in their final sorted position at the right end of the array.

This is an in-place algorithm because it uses only a constant time extra space besides that needed to store the array.

Unsorting

Start by swapping the rightmost element with a randomly chosen element from the left end of the array.

Decrement the swapped position and continue.

Since every time the element chosen is at random, the list will become sorted.

Two-Dimensional Arrays

- Declared as:

```
int[][] multTable = new int[rows][cols];
```

- Like all arrays, the size is fixed and must be set when the array is instantiated:

```
multTable = new int[9][9];
for (int i = 1; i < int.length; i++) {
    for (int j = 1; j < int[i].length; j++) {
            multTable[i][j] = i * j;
    }
}
```

12

Initializers for Two-Dimensional Arrays

- Declared as:

```
int[][] A = { { 1, 0,12,-1},
              { 7,-4,11,6},
              {-5,-2, 2,-3}
            };
```

- Separates the elements in each row by commas, and separates the rows by commas. Each row is in a pair of braces and the entire 2-D array is within a pair of enclosing braces.
- When processing a 2-D array, you should never assume that all the rows are of the same length. If you use the notation arr[i].length on the inside for loop you will be sure not to make an exception arise.

13

Two-Dimensional Arrays

- Not to be confused with parallel arrays.

  parallel arrays are arrays of the same size that are processed together.

"Ragged arrays" occur when each row of the array may have a unequal number of columns

```
triangularArray = new double[4][];
for (int i=0; i < 4; i++) {
    triangularArray = new double[i + 1];
}
```

14

Representation of Graphs

A graph G = (V, E) is a data structure composed of vertexes and edges. Each element of the vertex set V holds some unique key that distinguishes it from all others. The edges of the edge set E are specified by pairs, indicating the start and end vertex of each edge.

Graphs are used for many applications such as:

- Modeling computer networks, airplane routes, roads, neural pathways in the brain, etc.

But how can a computer recognize a graph?

15