**ITMD\_510 Week 3 & 4 Notes –1/24/18 Wed.**

**Arrays vs ArrayLists intro -Chapter 7**

* What are some uses?
* Data types
* Growth
* Sorting
* Searching
* Stacks (LIFO) / Queues (FIFO)
* List processing
* Know advantages / disadvantages of each.
* Compare and contrast each. *Know distinctions*!

Refs:

<https://www.tutorialspoint.com/java/java_arraylist_class.htm>

|  |  |  |
| --- | --- | --- |
| Features | Array | ArrayList  (from Collections framework) |
|  |  |  |
| Resizable? | **No**  **(fixed size)** | **Yes**  **(dynamic size)** |
|  |  |  |
| Primitives? | **Yes** | **No**  **(Reference based)** |
|  |  |  |
| Iterating values | **for, *for each*** | **iterator, *for each*** |
|  |  |  |
| Length | **length variable** | **size method** |
|  |  |  |
| Performance (speed)  -retrieving  -setting  -removal  -updating | **Fast** | **Slow in comparison** |
|  |  |  |
| Multidimensional? | **Yes** | **No** |
|  |  |  |
| Adding Elements | **assignment operator** | **add method** |
|  |  |  |
| Other methods… | **asList**  **sort**  **::** | **get**  **remove**  **::** |

Declarations

1. Array: Simple fixed sized arrays

**int** arr[] = new **int**[10]

**BankRecords** recs[] = new **BankRecords**[size]; //create array of objects!!

2. ArrayList: Dynamic sized arrays in Java that implement **List** *interface*.

ArrayList<**Type**> arrList = new ArrayList<**Type**>();

Here **Type** is a REFERENCE type to represent ArrayList elements to

be *created*

Notice above an Array is a fixed size data structure while ArrayList isn’t. ArrayList One need not to mention the size of ArrayList while creating its object. Even if we specify some initial capacity, we can add more elements.

Code comparisons

import java.util.ArrayList;

import java.util.Arrays;

**class Example {**

**public static void main(String args[]) {**

**/\* ........... Array............. \*/**

**int[] arr = new int[3];**

**arr[0] = 1;**

**arr[1] = 2;**

**arr[2] = 3;**

**// No more elements beyond here**

**/\*............ArrayList..............\*/**

**ArrayList<Integer> arrList = new ArrayList<Integer>();**

**arrList.add(1);**

**arrList.add(2);**

**arrList.add(3);**

**arrList.add(4);**

**// Able to continue to add more elements to arrList**

**System.out.println(arrList + “\n” + Arrays.toString(arr));**

**}**

**}**

-Fun Lil’ facts-

**There are 5 ways you can iterate through a List.**

1. For Loop
2. Advanced For Loop (***for each***)
3. While Loop
4. Iterator
5. Streams (Java8)