Agenda :

1. Continue discussing Collection framework
2. HashSet & TreeSet
3. DeQueue & PriorityQueue
4. LinkedList & ArrayList
5. Generics
6. Comparable & Comparator
7. Iterable & Iterator

Day 9 Revisit

1. Annotations – Annotations are metadata. It starts with @ symbol
2. Built-in annotation (@Deprecated, @Override, @SupressWarning, @FunctionalInterface etc.,) & Custom or User-defined annotations
3. Meta Annotations – Annotations used while creating custom annotation [@Target, @Documented, @Retention
4. Stack & Heap [both are memory location of JVM]
5. Stack is used to store ref. variables, program stacks, frame and heap is used to store Objects & string pools
6. Garbage Collection – Automatic Memory Management feature of Java. JVM will automatically reclaim the memory of unused objects. This is an automatic process that can’t be forced. JVM will do it internally. We may request JVM to garbage collect using
   1. System.gc()
   2. System.getRuntime.gc()
   3. System.runFinalize()
7. Collection API (List, Set & Queue – Interfaces)
8. Collection API is used to handle/manage group of objects.
9. Collection won’t work with primitive data
10. In case of primitive value passed to a collection, JVM will automatically convert them into objects using the wrapper class. This process is called as “AutoBoxing”

Always avoid using Pre-defined/Built-in Class names to the user-defined/custom class

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No | Valid Custom Class Name | Invalid Custom Class name | Reason |
| 1 | ArraysDemo | Arrays | Built-in class |
| 2 | ArrayListDemo | ArrayList | Built-in class |
| 3 | ListDemo | List | Built-in Interface |
| 4 | ThreadDemo | Thread | Built-in class |
| 5 | FileDemo | File | Built-in class |
| 6 | ScannerDemo | Scanner | Built-in class |

Bean Class is also called as model class or Entity Class or Data class

Generics – It’s a way of defining the elements type that will be stored in collections.

Package generally will be used for the following reasons

1. It is used to group similar type of classes
2. It is used to maintain multiple classes with the same name
3. It is also helpful in resolving the naming conflicts

Fully Qualified Class name – It means, representing a class along with it’s package name.

|  |  |  |
| --- | --- | --- |
| Sl No | Class Name | Fully Qualified Class/Interface Name |
| 1 | System | java.lang.System |
| 2 | List | java.util.List |
| 3 | Date | java.util.Date |
| 4 | Date | java.sql.Date |

Types of Classes

1. Simple Class / Concrete Class / Non-Abstract Class / Pojo Class
2. Abstract Class / Incomplete Class/ Non-Concrete class
3. Wrapper Classes
4. Bean Class & Entity Bean Class
5. Inner Class – A class defined inside another class
6. Anonymous class (Anonymous inner class) – Nameless class [Used to create object of abstract class]
7. Built-in or System/pre-defined class [ All the classes defined by Java are called Built-in classes]
8. Custom/User-defined classes [ All the classed developed by programmers are called Custom classes]
9. Base Class/Parent Class/Super Class [These class provides it’s public and protected members to other classes]
10. Derived Class/ Sub Class/Child Class [ These classes will get members from other Classes]

POJO – Plain Old Java Object – A class which is not extending other class nor implementing any interfaces

In Java, the filename and the public class name should be same.

In a single .java file, we can define n number of classes. But only one public class and that should be the name of the file

List<E> -- here E represents Element Type (It can be Wrapper Classes or Custom Classes)

HashSet VS TreeSet

HashSet

* Won’t maintain insertion order
* Allow null only once
* Duplicates are not allowed
* Fast Insertion & traversal

TreeSet

* It automatically sorts the element
* Duplicates are not allowed
* Insertion & removal of element will be slow
* Null value is not allowed bcos it can’t compared with any object for sorting

ArrayList VS LinkedList

ArrayList

* ArrayList is similar to Array but it can be resized dynamically (Dynamically sized Array)
* Traversing the element is fast bcos we can access the element randomly using the index value
* Insertion & removal will be slow bcos of shift left or shift right operation to fill the gap.
* List generally allows duplicates and maintains insertion order

LinkedList

* LinkedList implements both List & Queue interface (FIFO)
* Data will be stored in node
* Node will generally contain data and address of next node (Single Linked List)
* Node will generally contain data and address of previous &next node (doubly Linked List)
* Insertion & Deletion will be faster
* Traversing the element will be slower

ArrayDequeue VS PriorityQueue

ArrayDeQueue

* Stores the data in dynamically growable (resizable) array internally
* Implemented by both Queue & Stack
* It’s a implementation of double ended queue (so elements can be added/removed from either side)
* Imp methods push(), pop(), peekFirst(), peekLast(), pollFirst(), pollLast(),offerFirst(),offerLast()

PriorityQueue

* Generally Queue uses FIFO (First In First Out) algorithm
* But PriorityQueue will process few elements with high priority overriding the FIFO algorithm.
* Null values are not allowed bcos it can’t be compared with any other objects
* Can’t insert non comparable objects (Throws ClassCastException)
* Imp methods peek(), poll()

Iterable VS Iterator

Iterable

* Iterable Interface is defined in java.lang package
* It’s introduced in Java 5
* Is used to traverse using foreach loop
* Imp methods forEach(), iterator(),spliterator()

Iterator

* Iterator is an interface defined in java.util package
* It’s introduced in Java 1.2
* Imp methods hasNext(), next(), remove(), forEachRemaining()

Comparable Vs Comparator

Comparable

* It’s defined in java.lang package since java 1.2
* It is an interface used to do Natural ordering (comparision )
* Imp methods compareTo() -- lexicographical comparision
* It does the comparision of Strings based on it’s ASCII value

ASCII – American Standard Code for Information Interchange

Comparator

* It’s defined in java.util package since java 1.2
* It is an interface used to do define total ordering on collection
* Imp method is compare(T obj1, T obj2)

Generics

* Introduced in Java 5
* It enforce compile time safety (Only specified values are allowed in collection)
* Mainly used with collection
* E – Element – Used with all collection interfaces & classes
* K – Key --- used with Map
* V – Value -- used with Map

Generics is for compile time safety (Used with collection only) – To achieve homogeneous elements in collection.

Exception is for preventing/handling some un-expected situation. To provide smooth exit or to ignore certain lines of code when something un-expected happen.