Java Backend Development Live-54

lecture-2

Class Agenda

- Java OOPs Concepts: Abstraction
- Enumeration in Java
- Exception handling in Java
- Java Collections Framework
- Important Interfaces and their implementation.
- Practical use of these collections.

Final Classes and Methods

- Final methods cannot be overridden by subclasses
- Final class cannot be extended (or inherited) by any class.

Abstraction

- Data(implementation) hiding to reduce complexity.
- We don’t know how zoom is working, what is webRTC, We just want to use it.
- User focus on input and output not on implementation.
- Abstraction is achieved by abstract classes and interfaces.

Abstract classes cannot be instantiated, but they can be subclassed.

An abstract method is a method declared without implementation.

Subclass must implement all abstract method of parent class.

Abstract classes vs. Interfaces

Abstract class can have fields that are not static and final, and define public, protected and private concrete methods.

In interfaces all fields are automatically public, static, and final and all methods that you declare or define (as default method) are public.

A class can extend only one class, whether or not it is abstract, whereas it can implements any number of interfaces

Non-Access Modifiers

- Final
- Abstract
- Static
- Transient : skipped when serializing the object containing them.
- Synchronized: Methods can only accessed by one thread at a time.
- Volatile: The value of an attribute is not cached thread-locally, and is always read from “main memory”.

Enumeration in Java

- The Enumeration in Java is a way of defining a class with fixed and named constants.
- Java Enums are classes that have a fixed set of constants or variables that do not tend to change.
- Enum can be easily used in the switch. enum can be traversed. enum can have fields, constructors and methods.
- Enum improves **type safety** at compile-time checking to avoid errors at run-time by avoiding boilerplate code.

Example:

```
enum Day{

    SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY;

}
```

- Enum constants cannot be overridden
- Enum does not support Creation of Objects
- Enum cannot extend other classes
- Enum can implement Interface

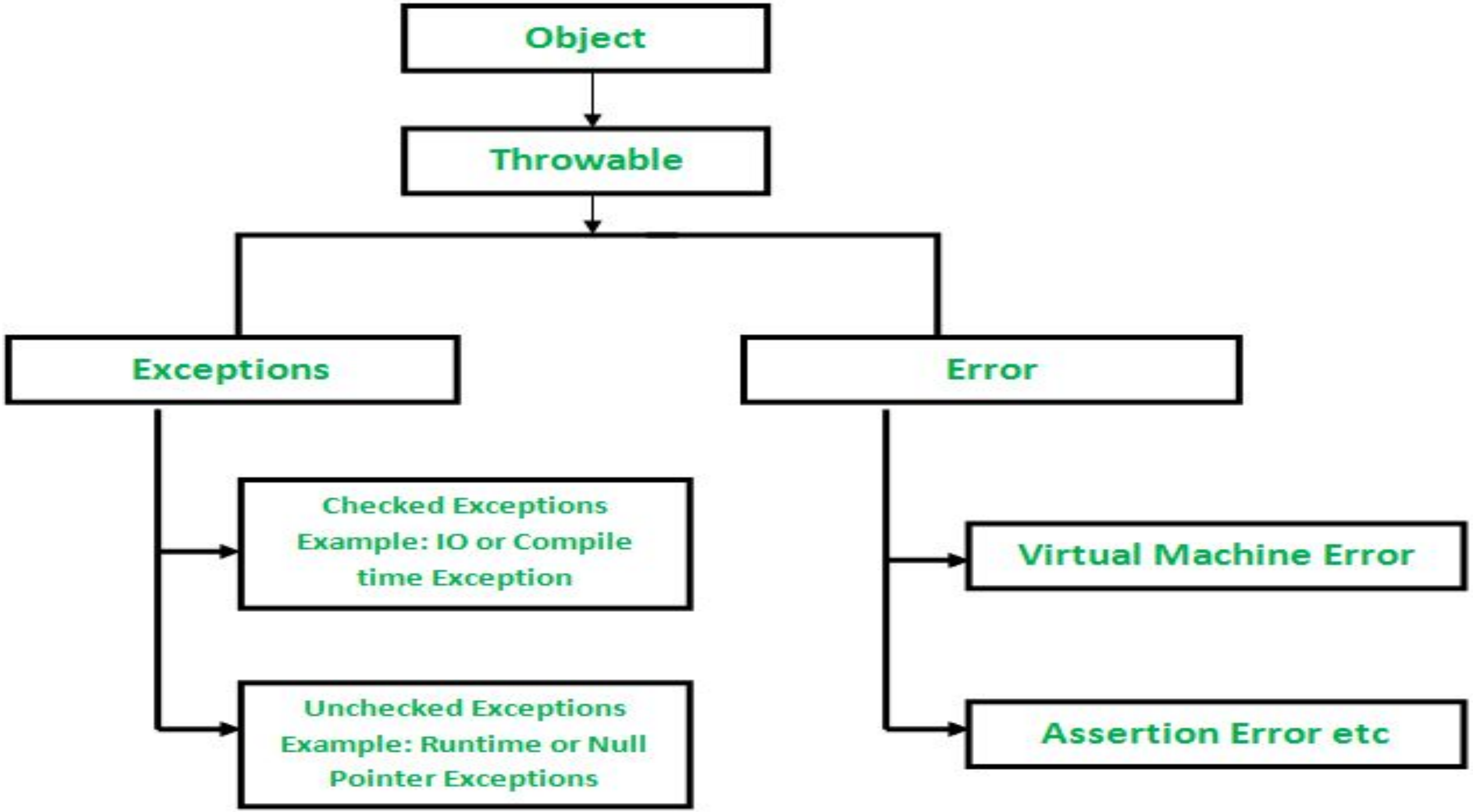
Exception

- unexpected/unwanted events.
- An exception is an event that disrupts the normal flow of the program.
- Leads to system failure.
- Exception handling is the mechanism to handle runtime errors such as Network, SQL, Invalid Data, FileNotFound etc.

Error vs. Exception

Error	**Exception**
Impossible to recover from error	Possible to recover from Exception
Errors are of Unchecked Type	It can be of checked or unchecked
Happen at Run-Time	Can happen at Compile time & Runtime
Caused by the environment on which the application is running	Caused by Application

Exception Hierarchy



Object

Throwable

Exceptions

Error

Checked Exceptions
Example: IO or Compile time Exception

Unchecked Exceptions
Example: Runtime or Null Pointer Exceptions

Virtual Machine Error

Assertion Error etc

Checked and Unchecked Exception

Checked Exception	**Unchecked Exception**
An exception that is checked by the compiler at compilation-time.	An exception that occurs at the time of execution.
These exceptions cannot simply be ignored, the programmer should handle these exceptions.	These are also called as Runtime Exceptions. Runtime exceptions are ignored at the time of compilation.
IOException, etc	NullPointerException, ArrayOurOfBoundException

Exception Handling Methods

Try: try block enclosed the code that can throw exception.

Catch: catch block provide exception handling code.

Multiple Catch: To handle multiple exceptions

Finally: finally block is always executed. It can be used to release resource acquired in try block.

Throw: keyword used to throw exceptions.

Throws: used to declare exceptions, used with method signature.

Final vs Finally vs Finalize

Final	Finally	Finalize
Keyword	Block	Method
Applies restrictions on class, method and variable.	Used to place an important code	Used to perform clean-up processing just before the object is garbage collected.
Final class cannot be inherited, method cannot be overridden & the variable value cannot be changed	It will be executed whether the exception is handled or not.	

User defined exception

- Built in exception: already available in JAVA

- We can define our own exception like ProductNotFoundException, BadRequestException, etc.

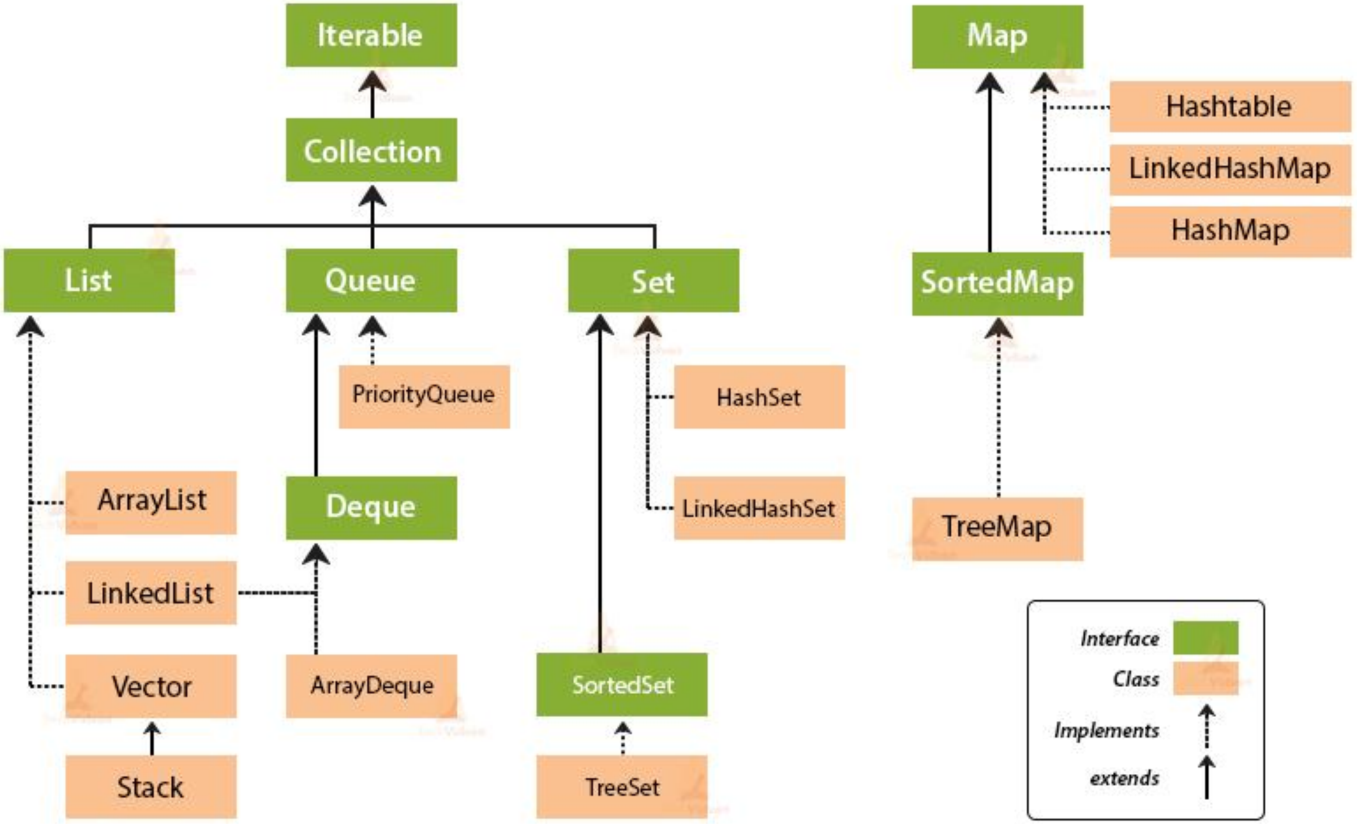
Java Collections Framework

- The Java language API provides many of the data structures from this framework for you.
- It defines a “collection” as “an object that represents a Group of elements (references to objects) It is not specified whether they are Ordered / not ordered Duplicated / not duplicated”.
- It defines a collections framework as “**a unified architecture for representing and manipulating collections,** allowing them to be manipulated independent of the details of their representation.”

Why Java Collection Framework ?

- Provides useful data structures and algorithms.
- Decreases extra effort required to learn, use, and design new API’s
- Supports reusability of standard data structures and algorithms

Collection Framework Hierarchy in Java



Iterator Interface

Iterator is an interface that iterates the elements. It is used to traverse the list and modify the elements.

- public boolean hasNext() – This method returns true if the iterator has more elements.
- public object next() – It returns the element and moves the cursor pointer to the next element.
- public void remove() – This method removes the last elements returned by the iterator.

List

A List is an ordered Collection of elements which may contain duplicates. It is an interface that extends the Collection interface. Lists are further classified into the following:

- ArrayList
- LinkedList
- Vectors

List implementations

ArrayList	**LinkedList**
Random Access: get(n) Constant time O(1)	Random Access: get(n) Linear time O(n)
Insert (beginning) and delete while iterating Linear time O(n)	Insert (beginning) and delete while iterating Constant time O(1)

Vectors: Similar to ArrayList but these are synchronized.

Practical uses of List

- Listing of product on amazon/flipkart.
- Listing of jobs on naukri.com
- Listing of questions of GeeksForGeeks

Queue

Queue in Java follows a FIFO approach i.e. it orders the elements in First In First Out manner. In a queue, the first element is removed first and last element is removed in the end.

Important methods:

- add()
- poll()
- peek()

Queue implementations

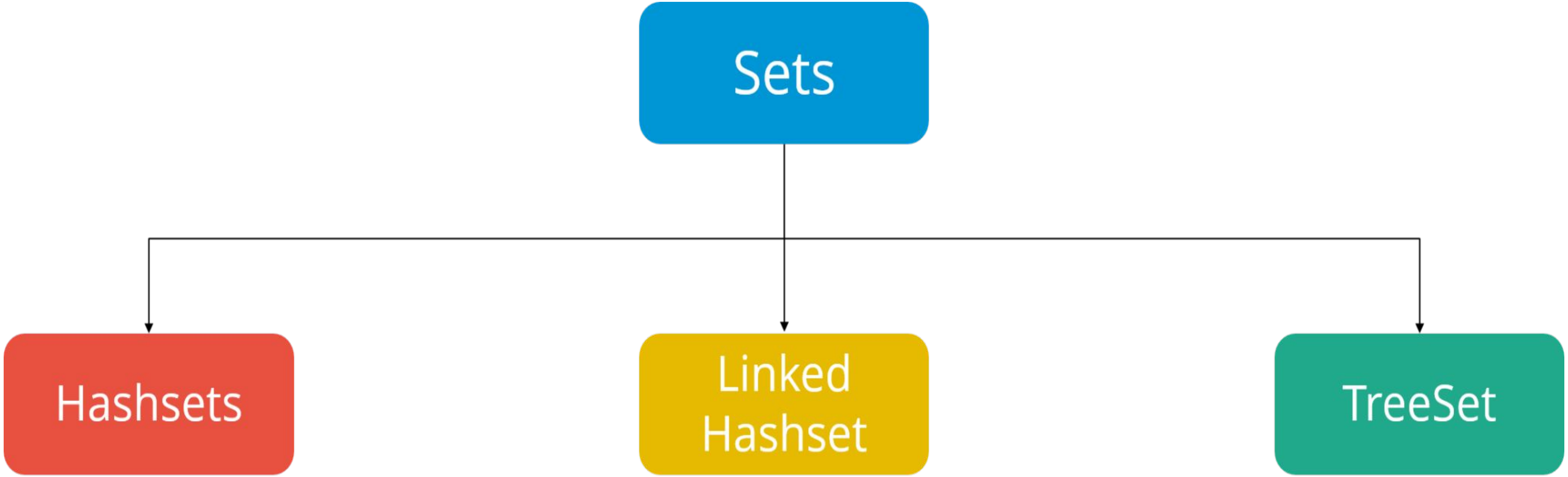
- **LinkedList**
  - head is the first element of the list
  - FIFO: Fist-In-First-Out
- **PriorityQueue**
  - The elements are ordered according to their natural ordering, or by a Comparator provided at the queue construction time

Practical uses of Queue

- Queue on ticket counter, waiting or RAC queue.
- Breadth First Search(BFS): Shortest distance between nodes.
- Level order traversal of tree.

Set Interface

A Set refers to a collection that cannot contain duplicate elements.



equals() and hashcode()

- equals() and hashCode() are **bound together by a joint contract** that specifies if two objects are considered equal using the equals() method, then they must have identical hashcode values.
- To be truly safe:
    - If override equals(), override hashCode()
    - Objects that are equals have to return identical hashcodes

Set implementations

- HashSet implements Set.
  - Hash tables as internal data structure (faster)
- LinkedHashSet extends HashSet
  - Elements are traversed by iterator according to the insertion order.
- TreeSet implements SortedSet.
  - R-B trees as internal data structure (computationally expensive)

Practical uses of Set

Find unique visitors.

Check username already exist.

Map Interface

- Data is stored in key-value pairs and every key is unique. Each key maps to a value hence the name map.
- Designed for the faster lookups.
- Analogous to Set.

Map implementations

- HashMap implements Map
  - No order
- LinkedHashMap extends HashMap
  - Insertion order
- TreeMap implements SortedMap
  - Ascending key order

Practical uses of Map

- Total hits on GeeksForGeeks country wise.
- API Rate Limiting.
- Find frequency of all char in a String.

Arrays and Collections class

Arrays and Collections classes provides several static methods that can be used to perform many tasks directly on arrays and collections.

- Fill an array/collection with a particular value.
- Sort an Arrays/Collections.
- Search in an Arrays/Collections.
- And many more.