Polymorphism  
  
runtime - overriding  
  
compile time - overloading

 Abstract classes

One abstract method should be there

Variables can be private  
cannot instantiate it  
method declarations and definations  
unimplemented methods must be implemented by the class which extends it (else we can declare the extend class also as abstract)

Interfaces  
  
Only abstract method declarations

Variables are final

Cannot be instantiated

Transient - the variables are not allowed in serialization

Storing the object into external file is called serialization and again reading it from it is called deserialization

Wrapper Classes: The primitive data types can be converted to object class

Garbage Collection :

Every object is created in heap area and string constant pool area(string literals are created here)

Daemon thread will scan the heap area and identify the object references which is idle for so much time, first it will mark all those objects and clean all of it.

System.gc() and runtime.gc() - explicit call to garbage

Final -

class level , - It cannot be inherited

|  |  |
| --- | --- |
|  | method level , it cannot be overrided |

Variable - constant

String - immutable(Once object is created, it's state cannot be changed)

Synchronization - The code should be made thread safe, execution process will be slowed by using this

Class level, method level, object level

Synchronized(a){

// list of statements will be executed at a time by each thread

}

Thread can be created by extending thread class or implementing runnable interface

Arrays vs Collections

Arrays are restricted to size and only specific datatypes are added, In collections we can add objects

Collections - Utility Class

Synchronized list is a method in the utility class, we can use this method to make your list as thread safe

List Iterator - to iterate through the elements in list

ArrayList - for retrieving the data

- ArrayList will implement random access interface so that indexing is done and we can access any element with same time

- Vector also implements Random access interface

- Marker Interface (Whenever our class is implementing the marker interface the class will have special behavior)

- For example a class can be cloneable if class implements cloneable Interface

Linked List- Underlying is double linked list

- map Insertion layer

- While deleting the elements

Vector - Synchronized methods (Other than that everything is same as arraylist)

- used for thread safe

- enumeration is used to iterate on vectors

- we can only walk through all the elements, cannot update the data and it will only iterate in the forward direction

HashSet - Underlying data structure is hash table

LinkedHashSet - Insertion order is preserved, underlying is linked hash set

Treeset - Underlying balanced tree, elements will be sorted order

- only first element can be inserted as NULL

- it will only allow same type of objects , it could not allow different types of objects since it had to compare two objects before inserting

collection - Interface

List Interface - Child interface of collection interface, duplicate elements are allowed and preserve the insertion order

* Underlaying data structure is array , so all the objects will be stored by indexing so it preserve the insertion order

Set Interface - Duplicate elements are not allowed and do not preserve the insertion order

- Hashing so insertion order is not preserved

Sorted Set- Child Interface of Set , all the elements are inserted in general sorting order

Navigable Set - Preserves Insertion order

Map Interface - It is not a child interface of collection

Sorted Map - The sorting of the keys

HashMap - An object in hashmap is not eligible for garbage collection , but in weak hashmap it is eligible for it

- All methods in hash table are synchronized

Comparable vs comparator interfaces

CompareTo method in comparable and compare method in comparator(it is used for customized sorting comparison)