Polymorphism .

Polymorphism in java is a concept by which we can perform a single action by different ways. Polymorphism is derived from 2 greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms.

Interface in Java

An interface in java is a blueprint of a class. It has static constants and abstract methods only.

The interface in java is a to achieve fully abstraction. There can be only abstract methods in the java interface not method body. It is used to achieve fully abstraction and multiple inheritance in Java.

Java Interface also represents IS-A relationship.

It cannot be instantiated just like abstract class.

Inheritance in Java

Inheritance in java is a mechanism in which one object acquires all the properties and behaviors of parent object.

The idea behind inheritance in java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of parent class, and you can add new methods and fields also.

Inheritance represents the IS-A relationship, also known as parent-child relationship.

Java Array

Normally, array is a collection of similar type of elements that have contiguous memory location.

Java array is an object the contains elements of similar data type. It is a data structure where we store similar elements. We can store only fixed set of elements in a java array.

Array in java is index based, first element of the array is stored at 0 index.

Java ArrayList class

Java ArrayList class uses a dynamic array for storing the elements.It extends AbstractList class and implements List interface.

Java ArrayList class can contain duplicate elements.

Java ArrayList class maintains insertion order.

Java ArrayList class is non synchronized.

Java ArrayList allows random access because array works at the index basis.

In Java ArrayList class, manipulation is slow because a lot of shifting needs to be occurred if any element is removed from the array list.

Collections in Java

Collections in java is a framework that provides an architecture to store and manipulate the group of objects.

All the operations that you perform on a data such as searching, sorting, insertion, manipulation, deletion etc. can be performed by Java Collections.

Java Collection simply means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque etc.) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet etc).

What is Collection in java

Collection represents a single unit of objects i.e. a group.

EJB (Enterprise Java Bean) is used to develop scalable, robust and secured enterprise applications in java.

Unlike RMI, middleware services such as security, transaction management etc. are provided by EJB Container to all EJB applications.

The current version of EJB is EJB 3.2. The development of EJB 3 is faster than EJB 2 because of simplicity and annotations such as @EJB,

Session Bean

Session bean encapsulates business logic only, it can be invoked by local, remote and webservice client.

It can be used for calculations, database access etc.

The life cycle of session bean is maintained by the application server (EJB Container).

Types of Session Bean

There are 3 types of session bean.

1) Stateless Session Bean: It doesn't maintain state of a client between multiple method calls.

2) Stateful Session Bean: It maintains state of a client across multiple requests.

3) Singleton Session Bean: One instance per application, it is shared between clients and supports concurrent access.

Servlet Tutorial

Servlet technology is used to create web application (resides at server side and generates dynamic web page).

Servet technology is robust and scalable as it uses the java language. Before Servlet, CGI (Common Gateway Interface) scripting language was used as a server-side programming language. But there were many disadvantages of this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse etc.

What is a Servlet?

Servlet can be described in many ways, depending on the context.

Servlet is a technology i.e. used to create web application.

Servlet is an API that provides many interfaces and classes including documentations.

Servlet is an interface that must be implemented for creating any servlet.

Servlet is a class that extend the capabilities of the servers and respond to the incoming request. It can respond to any type of requests.

Servlet is a web component that is deployed on the server to create dynamic web page.

iterator is used to get value stored in collection.

it has three methods.

1.hasNext()-return true if iterator has more element[retun false if next() throws exception)

2.next() - return the next element,it throw NoSuchElementException.

3.remove() -Removes from the underlying collection the last element returned by the iterator (optional operation). This method can be called only once per call to next.

In List we have get value using "get" but incase of Set there is no dafault method. So using iterator we get it.

for example.

Set<Integer> l=new TreeSet<Integer>();

l.add(2);

l.add(3);

Iterator<Integer> it=l.iterator();

while (it.hasNext()) {//check has more elements is present.If no more element is not present then loop exit.

System.out.println(it.next());//get the value,here print 2,3

}

ref:http://docs.oracle.com/javase/6/docs/api/java/util/Iterator.html

Java HashMap class

A HashMap contains values based on the key. It implements the Map interface and extends AbstractMap class.

It contains only unique elements.

It may have one null key and multiple null values.

It maintains no order.

Hierarchy of HashMap class:

Example of HashMap class:

import java.util.\*;

class TestCollection13{

public static void main(String args[]){

HashMap<Integer,String> hm=new HashMap<Integer,String>();

hm.put(100,"Amit");

hm.put(101,"Vijay");

hm.put(102,"Rahul");

for(Map.Entry m:hm.entrySet()){

System.out.println(m.getKey()+" "+m.getValue());

}

}

Java Hashtable class

A Hashtable is an array of list.Each list is known as a bucket.The position of bucket is identified by calling the hashcode() method.A Hashtable contains values based on the key. It implements the Map interface and extends Dictionary class.

It contains only unique elements.

It may have not have any null key or value.

It is synchronized.

Example of Hashtable:

import java.util.\*;

class TestCollection16{

public static void main(String args[]){

Hashtable<Integer,String> hm=new Hashtable<Integer,String>();

hm.put(100,"Amit");

hm.put(102,"Ravi");

hm.put(101,"Vijay");

hm.put(103,"Rahul");

for(Map.Entry m:hm.entrySet()){

System.out.println(m.getKey()+" "+m.getValue());

}

}

Local inner class

A class that is created inside a method is known as local inner class. If you want to invoke the methods of local inner class, you must instantiate this class inside the method.

Program of local inner class

public class localInner1{

private int data=30;//instance variable

void display(){

class Local{

void msg(){System.out.println(data);}

}

Local l=new Local();

l.msg();

}

public static void main(String args[]){

localInner1 obj=new localInner1();

obj.display();

}

Multithreading in Java

Multithreading in java is a process of executing multiple threads simultaneously.

Thread is basically a lightweight sub-process, a smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.

But we use multithreading than multiprocessing because threads share a common memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

Java Multithreading is mostly used in games, animation etc.

Advantage of Java Multithreading

1) It doesn't block the user because threads are independent and you can perform multiple operations at same time.

2) You can perform many operations together so it saves time.

3) Threads are independent so it doesn't affect other threads if exception occur in a single thread.

What is Thread in java

A thread is a lightweight sub process, a smallest unit of processing. It is a separate path of execution.

Threads are independent, if there occurs exception in one thread, it doesn't affect other threads. It shares a common memory area.

Life cycle of a Thread (Thread States)

A thread can be in one of the five states. According to sun, there is only 4 states in thread life cycle in java new, runnable, non-runnable and terminated. There is no running state.

But for better understanding the threads, we are explaining it in the 5 states.

The life cycle of the thread in java is controlled by JVM. The java thread states are as follows:

New

Runnable

Running

Non-Runnable (Blocked)

Terminated