POORNIMA

Experiment - 1

AIM: Implement the following Data structure in Java.

(a) Linked list (b) Stack

(c) Queues (d) Set (e) Map

Description: The java.util package contains all the classes and interfaces for collection framework.

Method for Collection interface →

There are many methods declared in the Collection interface. They are as follows-

	Method	Description
(1)	public boolean add (object element)	is used to insert an element in this collection
(2)	public boolean addAll (Collection c)	is used to insert the specified collection elements in the invoking collection.
(3)	public boolean remove (object element)	is used to delete the element from this collection.
(4)	public boolean removeAll (Collection c)	is used to delete all the elements of specified collection from the

POORNIMA

(5)	public boolean retainAll (Collection c)	is used to delete all the elements of invoking collection expect the specified collection.
(6)	public int size()	return the total number of elements in the collection
(7)	public void clear()	remove the total no. of element from the collection.
(8)	public boolean contains(object element)	is used to search an element.
(9)	public boolean containAll (Collection c)	is used to search the specified collection in this collection.
(10)	public Iterator iterator()	is return an iterator.
(11)	public Object [] toArray()	covert collection into Array
(12)	public boolean isEmpty()	checks if collection is empty.

POORNIMA

Skeleton of Java.Util.Collection framework →

```
public interface Collection <E> extends Iterable <E> {
    int size();
    boolean isEmpty();
    boolean contains(Object o);
    Iterator <E> iterator();
    Object[] toArray();
    <T> T[] toArray(T[] a);
    boolean add(E e);
    boolean remove(Object o);
    boolean addAll(Collection <? extends E> c);
    boolean removeAll(Collection <?> c);
    boolean retainAll(collection <?> c);
    void clear();
    boolean equals(Object o);
    int hashCode();
}
```

Algorithm for all collection Data structure →

Steps of creating collection:

(1) Create an object of Generic type E, T, K or V.
(2) Create a Model class or plain old java object (POJO) of type.
(3) Generate setters and getters
(4) Create collection object of type either Set or List or Map or Queue.

(5) Add Objects to the collections

Boolean add (E o)

(6) Add collection to the collection.

Boolean addAll (Collection)

(7) Remove or retain data from collection

Remove (Collection) retainAll (Collection)

(8) Iterate Objects using Enumeration or Iterator or ListIterator

Iterator listIterator ()

(9) Display Objects from collection ()

(10) END.

Sample Input

Sample Employee Data set :
(employee.txt)

e100, james, asst. prof, cse, 8000, 16000, 4000, 8.7
e101, jack, asst. prof, cse, 8350, 17000, 4500, 9.2
e102, jane, assc. prof, cse, 15000, 30000, 8000, 7.8
e104, john, prof., cse, 30000, 60000, 15000, 8.8
e105, peter, assoc. prof., cse, 16500, 33000, 8600, 6.9
e106, david, assoc. prof, cse, 18000, 36000, 9500, 8.3
e107, daniel, asst. prof, cse, 9400, 19000, 5000, 7.9
e108, ramu, assoc. prof, cse, 17000, 34000, 9000, 6.8
e109, rani, asst. prof, cse, 10000, 21500, 4800, 6.4
e110, murthy, prof., cse, 35000, 71500, 15000, 9.3

Expected Output

Prints all the information of employee with all its attributes

POORNIMA

Experiment -2

AIM: (i) Perform setting up and installing Hadoop in its three operating modes-

- Standalones
- Pseudo Distributed
- fully distributed

Description:

Hadoop is written in java, so you will need to installed java on your m/c version 6 or later. Sun's JDK is widly used for Hadoop.

Hadoop runs for ourself on windows Unix, linux is the only supported production platform, but other favour of Unix. to run Hadoop. Windows is only supported as a developed plateform and additionally requires Cygwin to run.

Algorithm:

Steps involved in installing Hadoop in standlone mode →

(1) Command for installing ssh is "sudo apt-get install ssh".

(2) Command for key generation is "ssh-keygen -t rsa -p".

(3) Store the key into rsa.pub by using the command cat $HOME/.ssh/id-rsa.pub >> $HOME/.ssh/authorzed-keys.

(4) Extract the java by using the command tar xvfz jdk-8u60-linux-i586.tar.gz.

POORNIMA

(5) Extract the eclipse by using command tar -xvfz eclipse-jee-mars-R-linux-gtk.tar.gz

(6) Extract the hadoop by using the command tar hadoop-2.7.1 tar.gz

(7) Move the java to /usr/lib/jvm and eclips to /opt/ paths. Configure the java path in the eclipse.ini file

(8) Export java path and hadoop path in ./bashrc

(9) Check the installation successfull.

(10) Check hadoop istance in standlone mode.

Algorithm →

Steps involved in installing Hadoop in psedo distributed mode →

(i) In order install pseudo distribution mode we need to configure the hadoop configuration files.

(ii) first configure the hadoop.env.sh file by changing java path.

(iii) Configure core-site.xml which contain property tag it contain name & value.

(iv) Configure hdfs-site.xml

(v) configure yarn-site.xml

(vi) Configure marped-site.xml file

(vii) Now format the name node by using command hdfs namenode -format.

POORNIMA

(viii) Type command start-dfs.sh , start-yarn.sh

(ix) Run JPS which view all daemons. Create directory and entre some data into lendi.txt and copy from local directory to hadoop using command.

(x) Display content of file by using command
hdfs dfs -cat /newdir/part-r-0000

Fully distributed mode installation →

Algorithm →

(1) Stop all nodes clusters
$ stop-all.sh

(2) Decide one as NameNode (master) and remaining as DataNode (Slaves).

(3) Copy public key to all three hosts to get a password less SSH Access.
$ ssh-copy-id -I $HOME/.ssh/id-rsa.pub lendi@15sys24

(4) Configure all configuration file -
$ cd $Hadoop_HOME/etc/hadoop
$ nano core-site.xml
$ nano hdfs-site.xml

(5) Add hostname to file slaves and save it.

$ nano slaves

(6) configure $ nano yarn-site.xml

(7) Do in Master Node

$ hdfs namenode -format

$ start-dfs.sh

$ start-yarn.sh

(8) formate namenode

(9) Daemon starting in Master & Slave

(10) END.

(II) Using Web based Tool to manage Hadoop set-up

Description

Hadoop setup can managed by different web based tools, which can be easy for the user to identify the running daemons.

(a) Apache Ambari

(b) Horton works

(c) Apache spark

INPUT →

Ubuntu @ localhost> jps

OUTPUT →

Datanode, namenodes Secondary namenode,
Node manager, Resource manager.

POORNIMA

Experiment-3

AIM : Implement the following file management tasks in Hadoop:

- Adding files & directories
- Retrieving files
- Deleting files

DESCRIPTION : HDFS is a scalable distributed file-system designed to scale to petabyte of data while running on top of the underlying filesystem of the OS. This allow Hadoop to efficiency schedule tasks to these node the contain data, or which are nearest to it. Hadoop provide the set of command line utilities that works similar to Linux file commands. We are going to look into HDFS file by interacting with it from the command line.

- Adding files and directories to HDFS
- Reteriving files from HDFS to local filesystem
- Deleting the files from HDFS

Algorithm

Syntax and command to add, reterive and delete the data from HDFS ⇒

POORNIMA

Step 1

Adding files and directories to HDFS

Before you can run Hadoop on data stored in HDFS, you will need to put the data into HDFS first. let's create directory. HDFS has a default working directory of /user/ $USER. for the purpose of illustration, we use chuck you should substitute your name in the example commands-

```
hadoop fs -mkdir /user/chuck
hadoop fs -put /example/test
hadoop fs - put example.txt /user/chuck.
```

Step 2

Reteriving files from HDFS

The Hadoop command get copies from the HDFS back to the local filesystem. To reterive example.txt we can run cmd-

```
hadoop fs - cat example.txt
```

Step 3

Delete files from HDFS

hadoop fs -rm example.txt

- command for creating a directory in hfds is "hdfs dfs- mkdir/ lendicse"
- Adding directory is done with command "hdfs dfs-put lend-english /".

Step 4

Coping data from NFS to HDFS

Copying data from directory command is "hdfs dfs-copy FromLocal".

- View the file by using the command "hdfs dfs -cat / lendi_english /glossary"
- Command for listing of items in Hadoop is "hdfs dfs -ls hdfs:// localhost:9000/"".
- Command for deleting files is "hdfs df -rm r/ kathreen"

INPUT

Input as any data format of type structured unstructured or semi- structured.

OUTPUT

Permission	owner	Group	Size	last modification	Block size	Name
drwxr-xr-x	lendi	Super-Group	0B	wed 17 Aug 2016 22:44:00 PM	0B	lendi_English
drwxr-xr-x	lendi	Super-Group	0B	wed 17 Aug 2016 23:10:00 AM	0B	Sandhana