**DNS SERVER TO RESOLVE A GIVEN HOST NAME**

Ex. No. : 312217205003

Date: P.G. ABINAYA

**PROGRAM**:

import java.net.InetAddress;

import java.net.\*;

import java.net.NetworkInterface;

import java.net.UnknownHostException;

import java.util.\*;

public class myip {

    public static void main(String args[]){

        try {

         System.out.println("Dns server to resolve a given host name");

         InetAddress host4 = InetAddress.getLocalHost();

         System.out.println("LocalHost IP : "+host4.getHostAddress()+"\nLocalHost Name : "+host4.getHostName());

            System.out.println("Enter the URL");

            Scanner a=new Scanner(System.in);

         String b=a.nextLine();

            InetAddress host2 = InetAddress.getByName(b);

            //InetAddress host5 = host2.getHostAddress();

            String c=host2.getHostAddress();

            System.out.println("Host IP : "+c+"\nHost Name : "+host2.getHostName());

            System.out.println("List of Ip addresses:");

            InetAddress[] host1 = InetAddress.getAllByName(b);

            for(InetAddress host:host1){

             System.out.println(host.getHostAddress());

            }

            char dd='.';

            int count=0;

            for(int i=0;i<c.length();i++)

            {

             if(c.charAt(i)==dd)

             {

             count++;

             }

            }

            System.out.print("Length of the Address:");

            if(count==3)

            {

             System.out.println("32-bit");

             System.out.println("It is a IPV4 address");

            }

            else if(count==5)

            {

             System.out.println("128-bit");

             System.out.println("It is a IPV6 address");

            }

            String arr=c.substring(0,c.indexOf('.'));

            int aa=Integer.parseInt(arr);

                if(aa>0 && aa<=127){

             System.out.println("Class A Unicast Address");}

             else if(aa>127 && aa<=191){

             System.out.println("Class B Unicast Address");}

             else if(aa>191 && aa<=223){

             System.out.println("Class C Unicast Address");}

else if(aa>223 && aa<=239){

             System.out.println("Class D Multicast Address");}

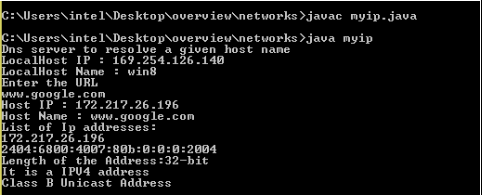
             else{

             System.out.println("Class E Reserved Address");}

                    } catch (UnknownHostException ex) {

            ex.printStackTrace();}}}

**SAMPLE INPUT/OUTPUT:**



**JAVA PROGRAMS USING RMI**

1. Greet the user and to display the local time.

Ex. No. : 312217205003

Date: P.G. ABINAYA

**PROGRAM**:

ClientRequest.java

import java.rmi.\*;

import java.util.Scanner;

import java.util.Calendar;

import java.util.GregorianCalendar;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Date;

public class ClientRequest{

    public static void main(String args[]) throws IOException{

     try{

     String[] answer=new String[5];

     String value=" ",answer1;

     int i=0,ch=1;

     Search access =(Search)Naming.lookup("rmi://localhost:1900"+ "/hel");

     do

     {

     i++;

answer[i] = access.query(value);

    System.out.println("Client :"+answer[i]);

    }while(!answer[i].equals("Exit"))} }

catch(Exception ae) {

System.out.println(ae);}} }

SearchServer.java

import java.rmi.\*;

import java.rmi.registry.\*;

public class SearchServer{

    public static void main(String args[]) {

try{

    Search obj = new SearchQuery();

LocateRegistry.createRegistry(1900);

Naming.rebind("rmi://localhost:1900"+

                          "/hel",obj); }

catch(Exception ae)  {

System.out.println(ae);

} } }

SearchQuery.java

import java.rmi.\*;

import java.rmi.server.\*;

import java.rmi.\*;

import java.io.\*;

import java.util.\*;

interface Search extends Remote

{

    public String query(String search) throws RemoteException;

}

public class SearchQuery extends UnicastRemoteObject

                         implements Search

{

   SearchQuery() throws RemoteException

    {

    }

    public String query(String search)

                       throws RemoteException

    {

        String[] res=new String[2];

        String d;

        String answer=" ";

        System.out.println("Enter the choice \n 1)greeting \n 2)Local Time \n 3)Exit");

    Scanner c=new Scanner(System.in);

    d=c.nextLine();

    switch(d){

        case "1":{

    System.out.println("Enter the user name");

    Scanner a=new Scanner(System.in);

    String b=a.nextLine();

        String value=b+"!!!!!!!!!!!!!!!!!!";

try{

            answer="Welcome "+value;

        }

catch(Exception ae) {

System.out.println(ae);}

break;

}

case "2":{

Calendar calendar = new GregorianCalendar();

    String am\_pm;

int hour = calendar.get(Calendar.HOUR);

int minute = calendar.get(Calendar.MINUTE);

int second = calendar.get(Calendar.SECOND);

    if(calendar.get(Calendar.AM\_PM) == 0)

am\_pm = "AM";

    else

am\_pm = "PM";

answer=hour+":"+minute+":"+second+" "+am\_pm;

    break;

}

case "3":

System.out.println("Exit!!!!!!!!!!!!!!!");

answer="Exit";

    break;

default:

System.out.println("Enter a valid choice");

}

System.out.println("Server :"+answer);

return answer;

    }

}

**SAMPLE INPUT/OUTPUT:**

