Experiment No. 6

Aim of the Experiment: Write a program to demonstrate the use of InetAddress class and its factory methods.

Objective:

To understand use of InetAddress class and its factory methods.

Resources: Eclipse IDE 2018, JDK 1.8.0 is required

Course Outcome Addressed: CO6

Theory:

Java InetAddress class represents an IP address. The java.net.InetAddress class provides methods to get the IP of any host name *for example* www.google.com, www.facebook.com, etc. An IP address is represented by 32-bit or 128-bit unsigned number. An instance of InetAddress represents the IP address with its corresponding host name.

There are two types of addresses: Unicast and Multicast.

The Unicast is an identifier for a single interface whereas Multicast is an identifier for a set of interfaces.

Moreover, InetAddress has a cache mechanism to store successful and unsuccessful host name resolutions.

The **InetAddress** class is used to encapsulate both the numerical IP address and the domain name for that address. You interact with this class by using the name of an IP host, which is more convenient and understandable than its IP address. The InetAddress class hides the number inside. InetAddress can handle both **IPv4** and **IPv6** addresses.

IP Address

- An IP address helps to identify a specific resource on the network using a numerical representation.
- Most networks combine IP with TCP (Transmission Control Protocol). It builds a virtual bridge among the destination and the source.

There are two versions of IP address:

1. IPv4

IPv4 is the primary Internet protocol. It is the first version of IP deployed for production in the ARAPNET in 1983. It is a widely used IP version to differentiate devices on network using an addressing scheme. A 32-bit addressing scheme is used to store 2^{32} addresses that is more than 4 million addresses.

Features of IPv4:

- It is a connectionless protocol.
- It utilizes less memory and the addresses can be remembered easily with the class based addressing scheme.
- It also offers video conferencing and libraries.

2. IPv6

IPv6 is the latest version of Internet protocol. It aims at fulfilling the need of more internet addresses. It provides solutions for the problems present in IPv4. It provides 128-bit address space that can be used to form a network of 340 undecillion unique IP addresses. IPv6 is also identified with a name IPng (Internet Protocol next generation).

Features of IPv6:

- It has a stateful and stateless both configurations.
- It provides support for quality of service (QoS).
- It has a hierarchical addressing and routing infrastructure.

TCP/IP Protocol

- TCP/IP is a communication protocol model used connect devices over a network via internet.
- TCP/IP helps in the process of addressing, transmitting, routing and receiving the data packets over the internet.
- The two main protocols used in this communication model are:
    1. TCP i.e. Transmission Control Protocol. TCP provides the way to create a communication channel across the network. It also helps in transmission of packets at sender end as well as receiver end.
    2. IP i.e. Internet Protocol. IP provides the address to the nodes connected on the internet. It uses a gateway computer to check whether the IP address is correct and the message is forwarded correctly or not.

InetAddress – Factory Methods :

The InetAddress class is used to encapsulate both, the numerical IP address and the domain name for that address. The InetAddress class has no visible constructors. The InetAddress class has the inability to create objects directly, hence factory methods are used for the purpose. The InetAddress class has no visible constructors. To create an InetAddress object, you have to use one of the available factory methods. Factory methods are merely a convention whereby static methods in a class return an instance of that class.

Factory Methods are static methods in a class that return an object of that class.

There are 5 factory methods available in InetAddress class –

Method	Description
static InetAddress getLocalHost() throws UnknownHostException	This method returns the instance of InetAddress containing the local hostname and address.
public static InetAddress getByName(String host) throws UnknownHostException	This method returns the instance of InetAddress containing LocalHost IP and name.

Method	Description
static InetAddress[] getAllByName(String hostName) throws UnknownHostException	This method returns the array of the instance of InetAddress class which contains IP addresses.
static InetAddress getByAddress(byte IPAddress[]) throws UnknownHostException	This method returns an InetAddress object created from the raw IP address.
static InetAddress getByAddress(String hostName, byte IPAddress[]) throws UnknownHostException	This method creates and returns an InetAddress based on the provided hostname and IP address.

Methods of InetAddress class:

Method	Description
public byte[] getAddress()	It returns the raw IP address of this InetAddress object as an array.
public String getHostAddress()	It returns IP address in textual form.
public boolean isAnyLocalAddress()	It returns true if this address represents a local address.
public boolean isLinkLocalAddress()	It returns true if this address is a link local address.
public boolean isLoopbackAddress()	It returns true if this address is a loopback address.
public boolean isMCGlobal()	It used to check whether the multicast address has a global scope or not
public boolean isMCLinkLocal()	It is used to check utility routine if the multicast address has a link scope or not.
public boolean isMCNodeLocal()	It is used to check the utility routine if the multicast address has node scope or not.
public boolean isMCOrgLocal()	It is used to check utility routine if the multicast address has

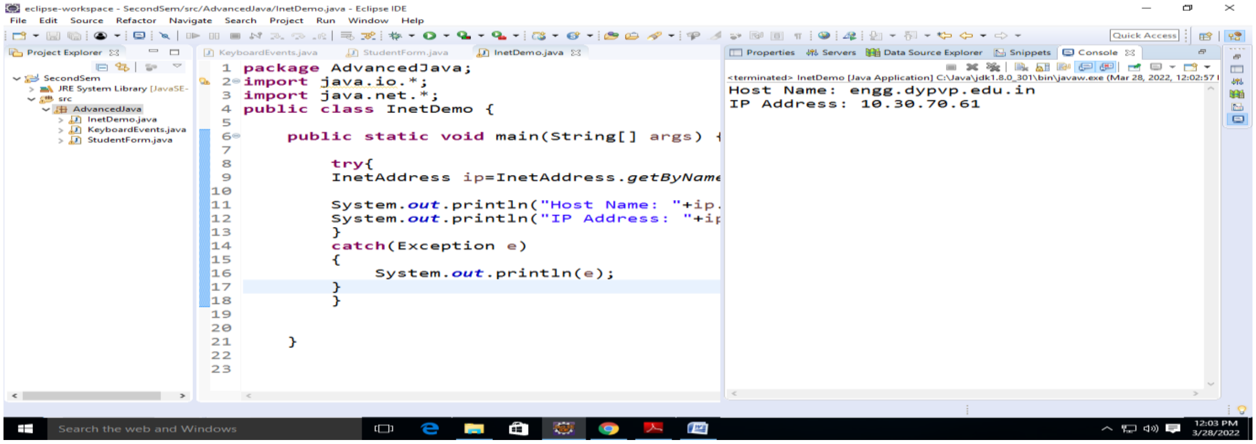
	organization scope or not.
public boolean isMCSiteLocal()	It returns true if this multicast address has site scope.
public boolean isMulticastAddress()	It returns true if this address is an IP multicast address.
public boolean isSiteLocalAddress()	It returns true if this address is a site local address.
public int hashCode()	It returns the hashcode associated with this address object.
public boolean equals(Object obj)	It returns true if this IP address is the same as that of the object specified.

SOURCE CODE:

Programme 1:

```
package AdvancedJava;
import java.io.*;
import java.net.*;
public class InetDemo {

public static void main(String[] args) {

try{
InetAddress ip=InetAddress.getByName("engg.dypvp.edu.in");

System.out.println("Host Name: "+ip.getHostName());
System.out.println("IP Address:"+ip.getHostAddress());
}
catch(Exception e)
          {
               System.out.println(e);
          }
}
}
```

OUTPUT:



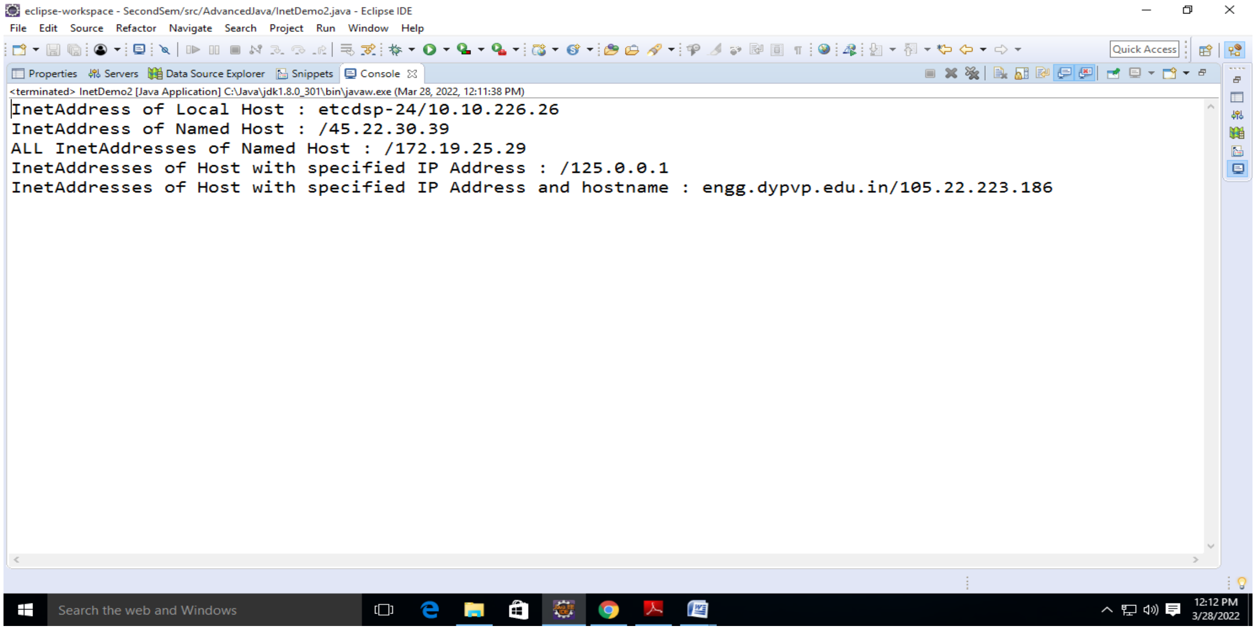
```
package AdvancedJava;
import java.io.*;
import java.net.*;
public class InetDemo {

    public static void main(String[] args) {

        try{
        InetAddress ip=InetAddress.getByName

        System.out.println("Host Name:  "+ip.
        System.out.println("IP Address:  "+ip
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        }


    }
```

```
<terminated> InetDemo [Java Application] C:\Java\jdk1.8.0_301\bin\javaw.exe (Mar 28, 2022, 12:02:57
Host Name: engg.dypvp.edu.in
IP Address: 10.30.70.61
```

Programme 2:

```
package AdvancedJava;
import java.io.*;
import java.net.*;
import java.util.*;

public class InetDemo2 {

public static void main(String[] args) throws UnknownHostException
 {
 // To get and print InetAddress of Local Host
InetAddress address1 = InetAddress.getLocalHost();
System.out.println("InetAddress of Local Host : " + address1);
    // To get and print InetAddress of Named Host
  InetAddress address2  = InetAddress.getByName("45.22.30.39");
System.out.println("InetAddress of Named Host : "   + address2);

  // To get and print ALL InetAddresses of Named Host
   InetAddress address3[]  = InetAddress.getAllByName("172.19.25.29");
for (int i = 0; i < address3.length; i++) {
System.out.println( "ALL InetAddresses of Named Host : "  + address3[i]);
}
// To get and print InetAddresses of
// Host with specified IP Address
 byte IPAddress[] = { 125, 0, 0, 1 };
 InetAddress address4  = InetAddress.getByAddress(IPAddress);
System.out.println("InetAddresses of Host with specified IP Address : " + address4);

 // To get and print InetAddresses of Host
 // with specified IP Address and hostname
byte[] IPAddress2  = { 105, 22, (byte)223, (byte)186 };
 InetAddress address5 = InetAddress.getByAddress( "engg.dypvp.edu.in", IPAddress2);
System.out.println("InetAddresses of Host with specified IP Address and hostname : "   + address5);
}
}
```

OUTPUT:



eclipse-workspace - SecondSem/src/AdvancedJava/InetDemo2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Properties Servers Data Source Explorer Snippets Console
Search the web and Windows
12:12 PM
3/28/2022

```
<terminated> InetDemo2 [Java Application] C:\Java\jdk1.8.0_301\bin\javaw.exe (Mar 28, 2022, 12:11:38 PM)
InetAddress of Local Host : etcdsp-24/10.10.226.26
InetAddress of Named Host : /45.22.30.39
ALL InetAddresses of Named Host : /172.19.25.29
InetAddresses of Host with specified IP Address : /125.0.0.1
InetAddresses of Host with specified IP Address and hostname : engg.dypvp.edu.in/105.22.223.186
```

Conclusion:

References:

Herbert Schildt , “Java : The Complete Reference” Tata McGraw-Hill (7th Edition).

Questions:

1. What are the purposes of using InetAddress class?
2. Which are the factory method of InetAddress class?
3. How are instances of InetAddress class created?
4. What happens if IP address of host Cannot be determined?
5. Which is the proper method to retrieve the host name of local machine?