

TÀI LIỆU THỰC TẬP LẬP TRÌNH MẠNG: LAB 07

DANH MỤC THUẬT NGỮ TIẾNG ANH

Từ	Nghĩa của từ
abstract	Trừu tượng
break	Dừng vòng lặp
catch	Từ khóa đầu của một khối bắt ngoại lệ
continue	Bỏ qua phần cuối vòng lặp, tiếp tục sang bước tiếp theo
default	Giá trị mặc định của phương thức switch()
extends	Kế thừa
final	Một hằng số, phương thức hay một lớp không được ghi đè
finally	Một phần của khối xử lý ngoại lệ try luôn được thực hiện
implements	Thực hiện giao diện
import	Khai báo một gói thư viện
instanceof	Kiểm tra một đối tượng là một thể hiện của lớp
interface	Giao diện
new	Tạo một đối tượng mới của lớp
null	Tham chiếu rỗng
package	Gói
private	Tiền tố chỉ được truy cập bởi phương thức của lớp
protected	Tiền tố được truy cập bởi phương thức của lớp, lớp con của và các lớp khác trong cùng một gói
public	Tiền tố có thể được truy cập bởi phương thức của tất cả các lớp
return	Trả về của một phương thức
super	Gọi phương thức của lớp cha
synchronized	Đồng bộ
this	Tham chiếu đến đối tượng hiện tại

DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
URL	Uniform Resource Locator
CSDL	Cơ Sở Dữ Liệu
JDBC	Java Database Connectivity
CNTT	Công Nghệ Thông Tin
HĐH	Hệ Điều Hành
MVC	Model-View-Control
DNS	Domain Name System
API	Application Programming Interface
FTP	File Transfer Protocol
JDK	Java Development Kit
GB	GigaByte
UCLN	Ước Chung Lớn Nhất
BCNN	Bội Chung Nhỏ Nhất
RAM	Random Access Memory
RMI	Remote Method Invocation
JVM	Java Virtual Machine
NIC	Network Interface Card
ĐH KTKT CN	Đại học Kinh tế Kỹ thuật Công nghiệp

LỜI NÓI ĐẦU

Ngày nay do nhu cầu thực tế và do sự phát triển mạnh mẽ của nhiều công nghệ tích hợp, dẫn đến các chương trình ứng dụng hầu hết đều có khả năng thực hiện trên môi trường mạng. Ngôn ngữ JAVA là ngôn ngữ phù hợp để viết các ứng dụng mạng. So với lập trình thông thường, lập trình mạng đòi hỏi người lập trình hiểu biết và có kỹ năng tốt để viết các chương trình giao tiếp và trao đổi dữ liệu giữa các máy tính với nhau.

Để hỗ trợ sinh viên chuyên ngành CNTT trong nhà trường tiếp cận với kỹ thuật lập trình mới này, tiếp theo cuốn tài liệu học tập lý thuyết “**Công nghệ JAVA**”, chúng tôi xây dựng cuốn “**Bài tập lập trình mạng**”, nhằm cung cấp cho sinh viên những kiến thức và kỹ thuật cơ bản nhất để phát triển các chương trình ứng dụng mạng, thông qua các dạng bài tập từ cơ bản đến nâng cao qua các chủ đề: lập trình cơ bản, lập trình hướng đối tượng, lập trình CSDL JDBC, lập trình mạng dùng socket, lập trình phân tán với RMI. Sinh viên sẽ thực hiện các bài thực hành này trên phòng máy nhà trường.

Nội dung cuốn tài liệu bao gồm 12 bài lab chia thành các chủ đề khác nhau. Trong mỗi chủ đề chúng tôi đưa ra tóm tắt lý thuyết, bài tập mẫu, sau đó là bài tập tương tự, và bài tập tổng hợp. Kết quả qua những bài lab, sinh viên được rèn và thành thạo các kỹ năng lập trình hướng đối tượng, lập trình CSDL, lập trình với giao thức truyền thông có sẵn và khả năng tích hợp trong các ứng dụng khác nhau, nhất là các giao thức truyền thông thời gian thực, từ đó sinh viên có thể viết được các phần mềm quản lý theo mô hình MVC, xây dựng được các ứng dụng mạng, các ứng dụng tích hợp và triệu gọi lẫn nhau trên mạng Intranet (mạng cục bộ), mạng Internet (mạng toàn cầu), các hệ thống xử lý truy xuất dữ liệu phân tán hoàn chỉnh. Nội dung biên soạn phù hợp với chuẩn đầu ra của ngành CNTT và ngành mạng máy tính và truyền thông dữ liệu về kỹ năng và kiến thức. Sau khi học xong học phần này sinh viên có thể viết phần mềm quản lý, truyền thông.

Chúng tôi xin chân thành cảm ơn Thầy Nguyễn Hoàng Chiến, phó chủ nhiệm khoa, phụ trách khoa CNTT trường ĐH KTKT CN cùng với các đồng nghiệp đã đóng góp ý kiến cho cuốn tài liệu này. Vì tài liệu được biên soạn lần đầu, chúng tôi đã cố gắng hoàn chỉnh, song không tránh khỏi thiếu sót. Rất mong nhận được sự góp ý của bạn đọc để tài liệu học tập được hoàn thiện hơn.

Xin trân trọng cảm ơn!

Nhóm tác giả

LAB 7. ĐỊA CHỈ IP, GIAO TIẾP MẠNG (NIC)[6]

A.MỤC TIÊU

- Trang bị cho sinh viên các thao tác lập trình với địa chỉ IP sử dụng lớp InetAddress nằm trong gói java.net, lập trình với giao tiếp mạng sử dụng lớp NetworkInterface.

B. NỘI DUNG

- Lập trình với InetAddress
- Lập trình hiển thị các giao tiếp mạng và tham số đi kèm.

C. KẾT QUẢ SAU KHI HOÀN THÀNH

- Sử dụng lớp InetAddress, sinh viên có thể tìm địa chỉ IP và hostname của máy cục bộ, lấy được địa chỉ của máy trạm thông qua tên miền cho trước.
- Sử dụng lớp NetworkInterface sinh viên có thể lấy được danh sách các giao tiếp mạng và các địa chỉ của chúng.

D. YÊU CẦU PHẦN CỨNG, PHẦN MỀM

- Máy tính cài HĐH windows, RAM tối thiểu 1GB, có kết nối Internet.
- Phần mềm NETBEAN 8.0, JDK 1.8.

E. HƯỚNG DẪN

1. Lớp InetAddress

Biểu diễn một địa chỉ Internet, bao gồm hai trường thông tin: hostName (một đối tượng kiểu **String**)- tên máy trạm và address (một số kiểu **int**)- địa chỉ IP của máy trạm đó.

Một số phương thức:

- getAddress() : Trả về địa chỉ IP chứa trong đối tượng InetAddress dạng mảng byte
- getAllByName(**String** host) :Trả về mảng địa chỉ của tất cả các máy trạm có cùng tên
- getHostName() : Trả về tên máy trạm chứa trong đối tượng.
- getLocalHost() : Trả về địa chỉ của máy cục bộ

Tạo đối tượng, sử dụng các phương thức tĩnh : getLocalHost(), getByName(**String**), getAllByName(**String**).

Tất cả các phương thức này đều thực hiện kết nối tới Server DNS cục bộ để biết được các thông tin trong đối tượng InetAddress.

Bài 1. In ra địa chỉ IP và hostname (tên máy trạm) của máy cục bộ

```
import java.net.*;
class myAddress
{
    public static void main (String args[])
    {
        try
```

```

        {
            InetAddress address = InetAddress.getLocalHost();
            System.out.println("Hello. My name is " +
address.getHostName() + " and my IP address is " +
address.getHostAddress());
        }
        catch (UnknownHostException e)
        {
            System.out.println("I don't know my own name and address.");
        }
    }
}

```

Bài 2. In ra địa chỉ các máy trạm trên mạng có cùng tên miền www.microsoft.com.

```

import java.net.*;
public class AllAddr
{
    public static void main(String[]args)
    {
        try
        {
            InetAddress[] addr = InetAddress.getAllByName("www.microsoft.com");
            for (int i = 0; i < addr.length; i++)
            {
                System.out.println (addr[i]);
            }
        }
        catch (UnknownHostException ex)
        {
            System.out.println("Could not find www.microsoft.com");
        }
    }
}

```

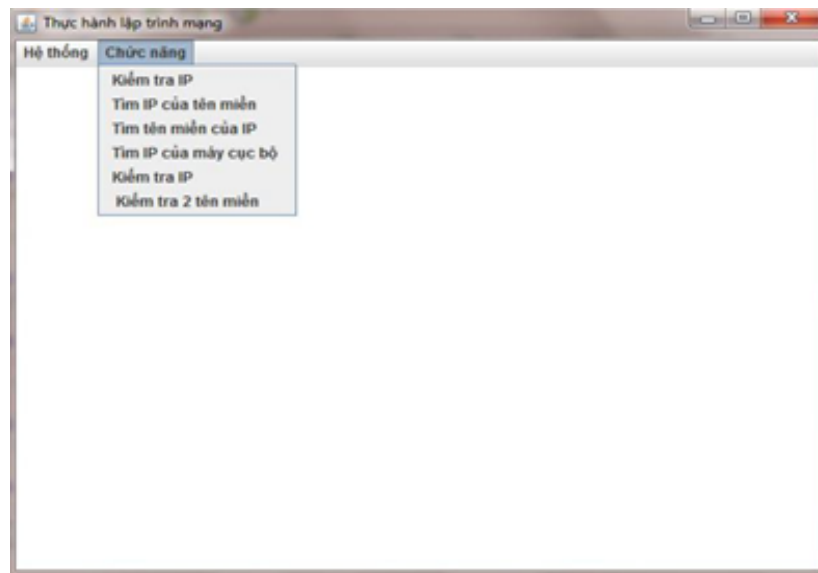
Dịch chạy chương trình trên máy tính có kết nối mạng Internet, kết quả trả về như sau:

```

www.microsoft.com/63.211.66.123
www.microsoft.com/63.211.66.124
www.microsoft.com/63.211.66.131
www.microsoft.com/63.211.66.11

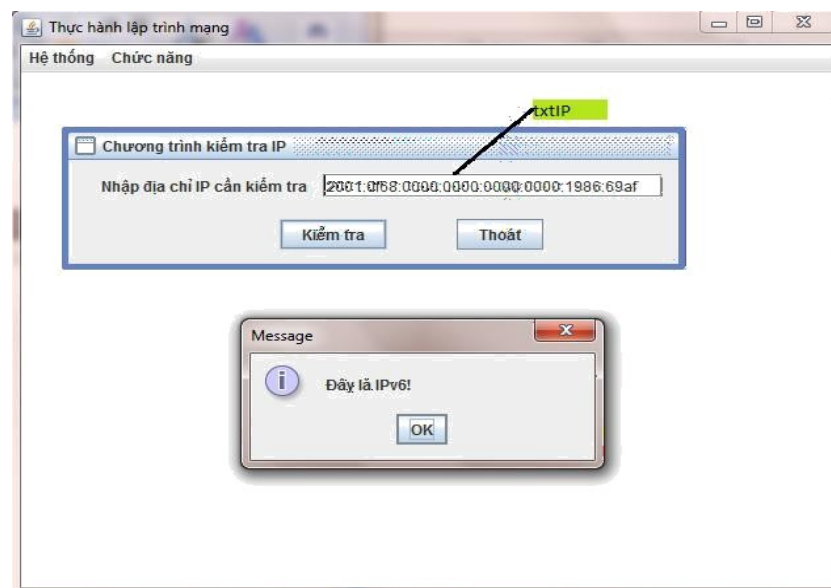
```

Bài 3. Tạo giao diện như hình sau: Thiết kế các chức năng tương ứng



a. Kiểm tra một tên miền có địa chỉ IP là phiên bản 4 hay phiên bản 6.

Bước 1: Tạo JFrameForm có giao diện như sau:



Bước 2: Xử lý sự kiện cho button Kiểm tra

```
private void btnKiemTraActionPerformed(java.awt.event.ActionEvent evt) {
    String IP = txtIP.getText();
    try
    {
        InetAddress host = InetAddress.getByName(IP);
        if(host != null)
        {
            if(IP.contains("."))
```

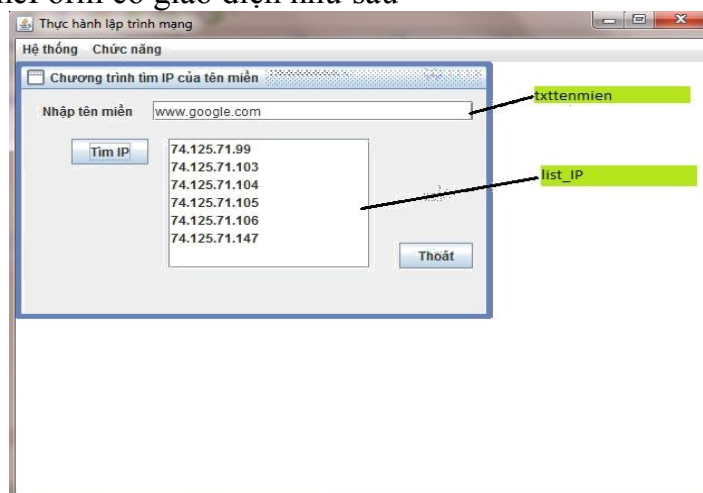
```

        JOptionPane.showMessageDialog(null, "Đây là IPv4");
    }
    else
        JOptionPane.showMessageDialog(null, "Đây là IPv6");
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Địa chỉ IP nhập sai!!!");
    }
}
catch (UnknownHostException ex)
{
    JOptionPane.showMessageDialog(null, "Địa chỉ IP nhập sai:
\n" + ex.toString());
}
}

```

b. Nhập địa chỉ tên miền. Cho biết tất cả địa chỉ IP của tên miền tương ứng.

Bước 1: Tạo JFrameForm có giao diện như sau



Bước 2: Xử lý sự kiện cho button tìm IP

```

private void btnTimIPActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
        int i, j;
        InetAddress addr[] =
InetAddress.getAllByName(txtTenMien.getText());
        DefaultListModel dlm = new DefaultListModel();
    }
}

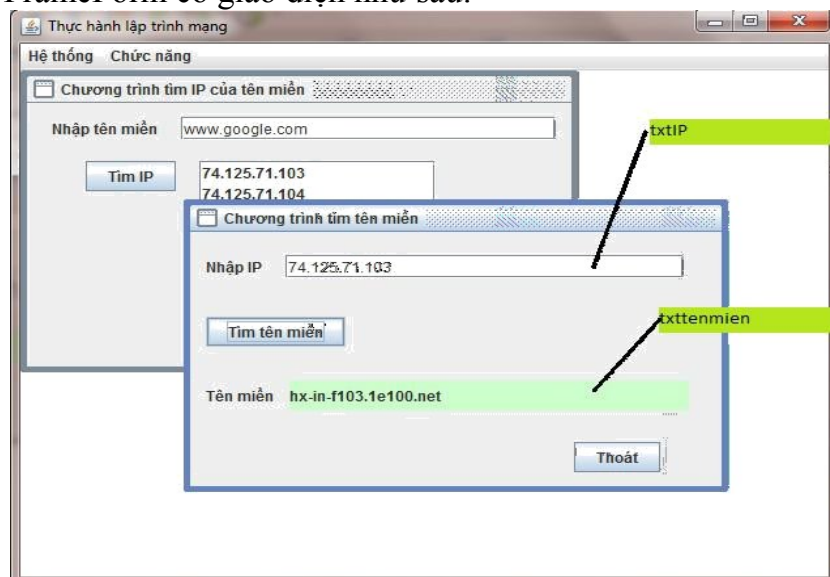
```

```

        for (i = 0; i < addr.length; i++)
        {
            byte[] ipAddr = addr[i].getAddress();
            String ipAddrStr = "";
            for (j = 0; j < ipAddr.length; j++)
            {
                if(j > 0)
                    ipAddrStr += ".";
                ipAddrStr += ipAddr[j]&0xFF;
            }
            dlm.addElement(ipAddrStr);
        }
        list_IP.setModel(dlm);
    }
    catch(UnknownHostException e)
    {
        JOptionPane.showMessageDialog(null, "Địa chỉ nhập sai!!!");
    }
}

```

c. Viết chương trình nhập vào địa chỉ IP của một máy. Cho biết tên miền tương ứng.
Bước 1: Tạo JFrameForm có giao diện như sau:



Bước 2: Xử lý sự kiện cho button tìm tên miền.

```

private void btnTimTenMienActionPerformed(java.awt.event.ActionEvent evt)
{
    try
    {
        InetAddress addr = InetAddress.getByName(txtIP.getText());
        String hostname = addr.getHostName();
    }
}

```



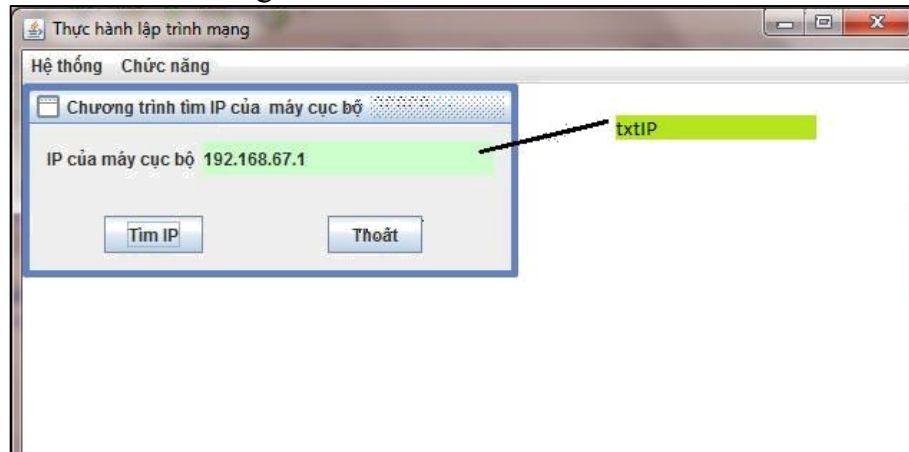
```

        txtTenMien.setText(hostname);
    }
    catch (UnknownHostException ex)
    {
        JOptionPane.showMessageDialog(null, "Bạn nhập sai tên miền");
    }
}

```

d. Viết chương trình cho biết địa chỉ IP của máy cục bộ.

Bước 1: Tạo JFrameForm có giao diện như sau:



Bước 2: Xử lý sự kiện cho button Tìm IP.

```

private void btnTimIPCucBoActionPerformed(java.awt.event.ActionEvent evt)
{
    try
    {
        InetAddress addr = InetAddress.getLocalHost();
        byte[] ipAddr = addr.getAddress();
        String ipAddrStr = "";
        for (int i = 0; i < ipAddr.length; i++)
        {
            if(i > 0)
                ipAddrStr += ".";
            ipAddrStr += ipAddr[i]&0xFF;
        }
        txtIP.setText(ipAddrStr);
    }
    catch (UnknownHostException ex)
    {
        JOptionPane.showMessageDialog(null, "Lỗi!!!");
    }
}

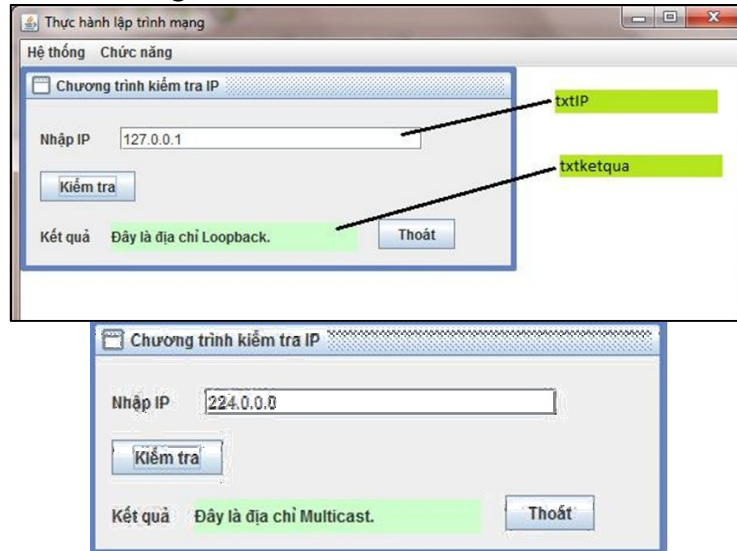
```

```
}
```

```
}
```

e. Nhập địa chỉ IP, kiểm tra đặc điểm của địa chỉ IP này có là một trong các dạng sau: địa chỉ cục bộ, địa chỉ loopback, địa chỉ multicast.

Bước 1: Tạo JFrameForm có giao diện như sau:



Bước 2: Xử lý sự kiện cho button Kiểm tra.

```
private void btnKTDacDiemIPActionPerformed(java.awt.event.ActionEvent
evt)
{
    try
    {
        InetAddress add = InetAddress.getByName(txtIP.getText());
        InetAddress localhost = InetAddress.getLocalHost();
        if(add.equals(localhost))
        {
            txtKetQua.setText("Đây là địa chỉ localhost");
        }
        else if(add.isMulticastAddress())
        {
            txtKetQua.setText("Đây là địa chỉ Multicast");
        }
        else if(add.isLoopbackAddress())
        {
            txtKetQua.setText("Đây là địa chỉ Loopback");
        }
        else
        {
            txtKetQua.setText("Không thấy gì đặc biệt");
        }
    }
}
```

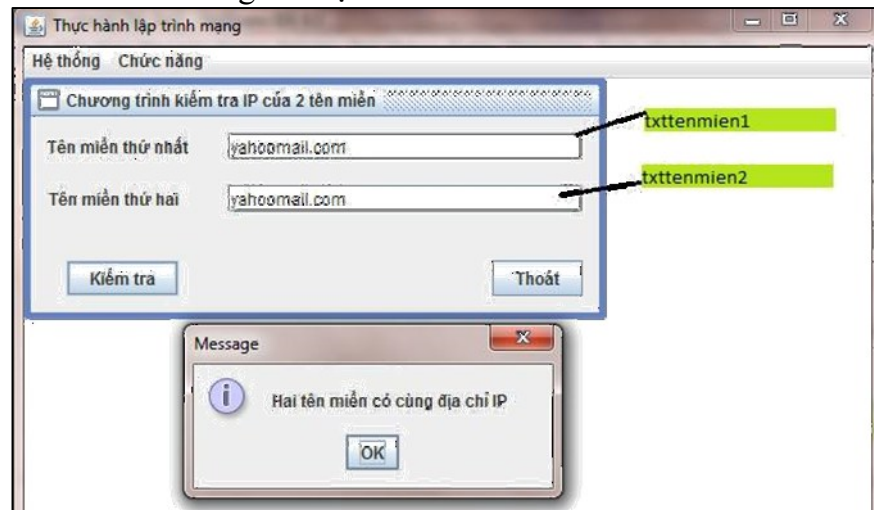
```

    }
}
catch (UnknownHostException ex)
{
}
}

```

f. Viết chương trình nhập vào 2 địa chỉ tên miền khác nhau. Kiểm tra xem hai tên miền này có cùng địa chỉ IP không?

Bước 1: Tạo JFrameForm có giao diện như sau:



Bước 2: Xử lý sự kiện cho button Kiểm tra.

```

private void btnKiemTraCungIPActionPerformed(java.awt.event.ActionEvent
evt)
{
    InetAddress add1[], add2[];
    try
    {
        add1 = InetAddress.getAllByName(txtTenMien1.getText());
        add2 = InetAddress.getAllByName(txtTenMien2.getText());
        if(Arrays.equals(add1, add2))
            JOptionPane.showMessageDialog(null, "Hai tên miền cùng địa chỉ IP");
        Else
            JOptionPane.showMessageDialog(null, "Hai tên miền không cùng địa chỉ IP");
    }
    catch(UnknownHostException ex)
    {
        JOptionPane.showMessageDialog(null, ex.toString());
    }
}

```

2. Lớp NetworkInterface

Cung cấp các phương thức để liệt kê tất cả các địa chỉ cục bộ và tạo ra đối tượng InetAddress.

getByName():

Trả về đối tượng NetworkInterface biểu diễn một bộ giao tiếp mạng với tên cụ thể.

getByInetAddress():

Trả về đối tượng NetworkInterface biểu diễn giao tiếp mạng được gắn với một địa chỉ IP cụ thể.

getNetworkInterfaces():

Trả về đối tượng java.util Enumeration là một danh sách liệt kê tất cả các giao tiếp mạng có trên máy cục bộ

Bài 4 : Viết chương trình hiện danh sách tất cả các giao tiếp mạng trên máy cục bộ.

```
public class InterfaceLister {
    public static void main(String[] args) throws Exception {
        Enumeration interfaces = NetworkInterface.getNetworkInterfaces( );
        while (interfaces.hasMoreElements( )) {
            NetworkInterface ni = (NetworkInterface) interfaces.nextElement(
                );
            System.out.println(ni);
        }
    }
}
```

Bài 5: Viết chương trình hiển thị tên của tất cả các giao tiếp mạng và giao tiếp con (nếu nó tồn tại) trên một máy.

```
public class ListNIFs
{
    public static void main(String args[]) throws SocketException {
        Enumeration<NetworkInterface> nets =
            NetworkInterface.getNetworkInterfaces();
        for (NetworkInterface netIf : Collections.list(nets)) {
            out.printf("Display name: %s\n", netIf.getDisplayName());
            out.printf("Name: %s\n", netIf.getName());
            displaySubInterfaces(netIf);
            out.printf("\n");
        }
    }

    static void displaySubInterfaces(NetworkInterface netIf) throws
        SocketException {
        Enumeration<NetworkInterface> subIfs = netIf.getSubInterfaces();
        for (NetworkInterface subIf : Collections.list(subIfs)) {
            out.printf("\tSub Interface Display name: %s\n",
```

```

        subIf.getDisplayName());
        out.printf("\tSub Interface Name: %s\n", subIf.getName());
    }
}
}

```

Truy cập các tham số giao tiếp mạng

Người sử dụng có thể truy cập các tham số về giao tiếp mạng ngoài tên và địa chỉ IP gán cho nó. Chương trình có thể phát hiện giao tiếp mạng đang chạy với phương thức `isUp()`. Các phương thức sau chỉ thị kiểu giao tiếp mạng:

isLoopback(): chỉ thị giao tiếp mạng là một giao tiếp loopback.

isPointToPoint(): chỉ thị nếu giao tiếp là giao tiếp point-to-point.

isVirtual(): chỉ thị nếu giao tiếp là giao tiếp ảo (giao tiếp mềm).

supportsMulticast(): chỉ thị khi giao tiếp mạng hỗ trợ multicast.

getHardwareAddress(): trả về địa chỉ phần cứng vật lý của giao tiếp mạng, địa chỉ MAC

getMTU(): trả về đơn vị truyền cực đại(MTU) là kích cỡ gói tin lớn nhất.

Bài 6:

Viết chương trình hiển thị danh sách các giao tiếp mạng và các tham số đi kèm.

```

public static void main(String args[]) throws SocketException {
    Enumeration<NetworkInterface> nets
    =NetworkInterface.getNetworkInterfaces();
    for (NetworkInterface netint : Collections.list(nets))
        displayInterfaceInformation(netint);
    }
    static void displayInterfaceInformation(NetworkInterface netint) throws
        SocketException {
        out.printf("Display name: %s \n", netint.getDisplayName());
        out.printf("Name: %s\n", netint.getName());
        Enumeration<InetAddress> inetAddresses = netint.getInetAddresses();
        for (InetAddress inetAddress : Collections.list(inetAddresses)) {
            out.printf("InetAddress: %s\n", inetAddress);
        }
        out.printf("Up? %s\n", netint.isUp());
        out.printf("Loopback? %s\n", netint.isLoopback());
        out.printf("PointToPoint? %s\n", netint.isPointToPoint());
        out.printf("Supports multicast? %s\n", netint.supportsMulticast());
        out.printf("Virtual? %s\n", netint.isVirtual());
        out.printf("Hardware address:
%s\n",Arrays.toString(netint.getHardwareAddress()));
        out.printf("MTU: %s\n", netint.getMTU());
    }
}

```

```
out.printf("\n");  
    }  
}  
}
```

TÀI LIỆU THAM KHẢO

- [1]. Cay S. Horstmann, *Core Java Volum I - Fundamentals, Tenth Edition*, NewYork : Prentice Hall, 2016.
- [2]. Cay S. Horstmann. *Core Java Volum II - Advanced Features, Tenth Edition*, New York : Prentice Hall, 2017.
- [3].Eng.haneen Ei-masry, *Java database connection*, Islamic University of Gaza Faculty of Engineering Department of Computer Engineering ECOM 4113: DataBase Lab, 2014.
- [4]. Angelos Stavrou, *Advanced Network Programming Lab using Java*, Network Security, ISA 656, Angelos Stavrou.
- [5]. Marenglen Biba, Ph.D, *Manual for Lab practices, Remote Method Invocation Three Tier Application with a Database Server*, Department of Comsputer Science, University of New York.
- [6].Elliotte Rusty Harold, *Java Network Programming, Fourth Edition*, O'Reilly Media, 2013.
- [7]. Đoàn Văn Ban, *Lập trình hướng đối tượng với JAVA*, Nhà xuất bản Khoa học và Kỹ thuật, 2005.
- [8]. ThS. Dương Thành Phết, *Bài tập thực hành Chuyên đề 1 CNPM- Java*, Khoa CNTT- Trường ĐH Công nghệ TP.HCM.
- [9]. <https://www.oracle.com/technetwork/java/socket-140484.html#>
- [10]. https://personales.unican.es/corcuerp/java/Labs/LAB_22.htm
- [11]. <http://www.nrcmec.org/pdf/Manuals/CSE/student/2-2%20java16-17.pdf>
- [12]. <http://cse.mait.ac.in/pdf/LAB%20MANUAL/JAVA.pdf>
- [13]. https://www.academia.edu/35283541/Bài_tập_môn_lập_trình_hướng_đối_tượng