



Mạng máy tính căn bản

Bài thực hành số 03

Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows

Họ tên sinh viên:

MSSV:.....

I. Mục tiêu:

- Thiết lập được môi trường lập trình Java trên Windows
- Nắm được một số kỹ năng cơ bản về lập trình Java
- Sử dụng được công cụ Eclipse
- Lập trình socket

II. Nội dung

1. Cài đặt JDK trong Windows

- Vào trang: <http://www.oracle.com/technetwork/java/javase/downloads/index.html> để tải JSK

Java SE Downloads

[Latest Release](#) [Next Release \(Early Access\)](#) [Embedded Use](#) [Previous Releases](#)


DOWNLOAD 


DOWNLOAD 


DOWNLOAD 

Java Platform (JDK) 7u21 JavaFX 2.2.21 JDK 7 + NetBeans

- Chọn Java Platform (JDK). Kết quả ta sẽ thấy các thông tin sau:

Solaris SPARC	95.44 MB	 jdk-7u21-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	22.97 MB	 jdk-7u21-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.58 MB	 jdk-7u21-solaris-sparcv9.tar.gz
Windows x86	88.98 MB	 jdk-7u21-windows-i586.exe
Windows x64	90.57 MB	 jdk-7u21-windows-x64.exe

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows

- Chọn package phù hợp với các hệ điều hành của mình. Ví dụ: ở đây ta chọn jdk-7u21-windows-i586.exe vì máy tính đang chạy Windows 32bit.

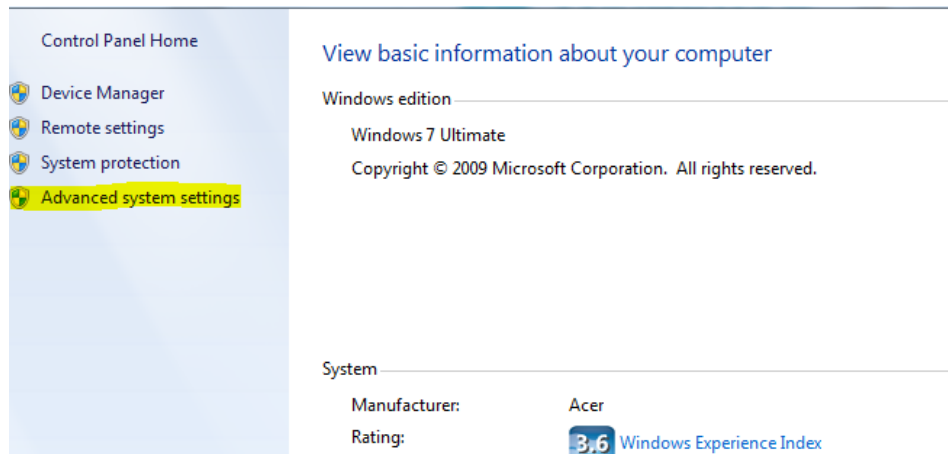
- Tiến hành cài đặt. Việc cài đặt này giống như cài đặt một chương trình bình thường.

2. Thiết lập biến môi trường để chạy Java trong command line trên môi trường window

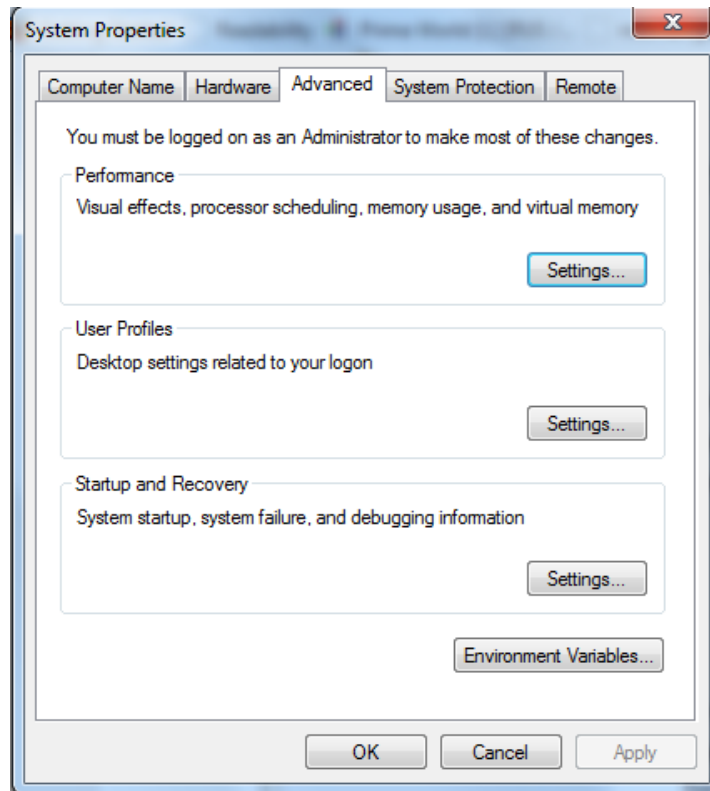
Lưu ý: môi trường thiết lập ở đây là window 7. Nếu các bạn sử dụng hệ điều hành khác hoặc window 10 thì các bước cài đặt ở đây có thể không theo các hình vẽ dưới đây. Nhưng về cơ bản chúng ta phải thiết lập môi trường chạy Java trong system properties.

Công việc này giúp chúng ta có thể thao tác với Java mà không cần dùng tới các IDE (Eclipse, Netbean,...). Ta có thể lập trình trên notepad hoặc notepad++ và chạy bằng cmd. Thích hợp cho việc làm quen code trong những ngày đầu làm quen với lập trình Java.

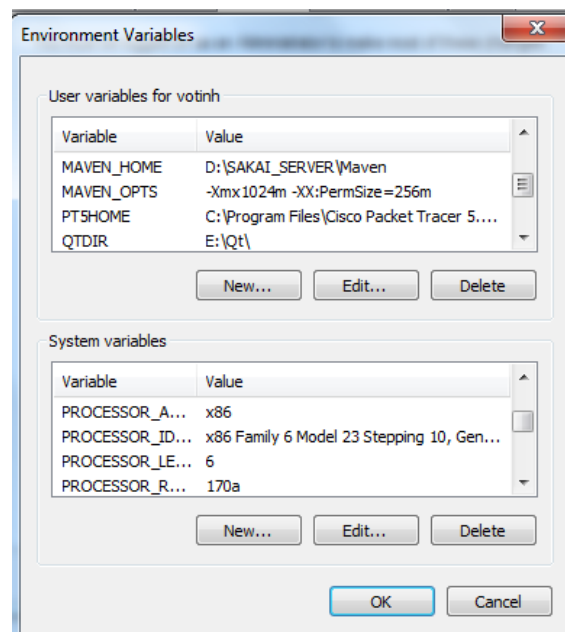
- Bước 1: Click phải vào **My computer** chọn **properties** sẽ xuất hiện cửa sổ:



- Bước 2: Chọn **Advanced system settings** sẽ xuất hiện cửa sổ **System Properties**



- Bước 3: Trong **tab Advanced** chọn **Environment Variables** sẽ xuất hiện **cửa sổ Environment Variables** như sau:



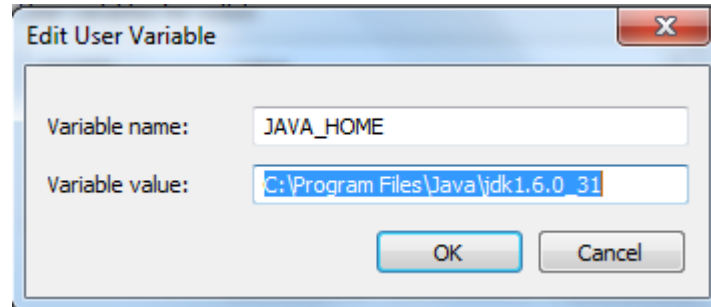
- Có 2 loại biến môi trường:

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows

+ **JAVA_HOME**: Chỉ đường dẫn đến nơi cài đặt JDK

+ **Path**: Chứa danh sách các thư mục mà chương trình sẽ tìm kiếm cho file thực thi tương ứng với tên lệnh được đưa ra bởi người dùng.

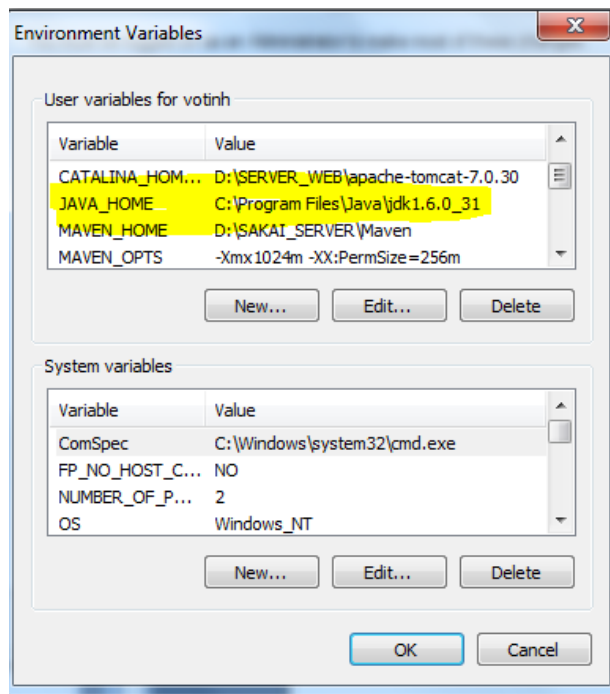
- Bước 4: Trong mục **System variables** chọn **New...** sẽ xuất hiện cửa sổ **New System Variables**



Trong khung giá trị Variable name, nhập vào biến **JAVA_HOME**

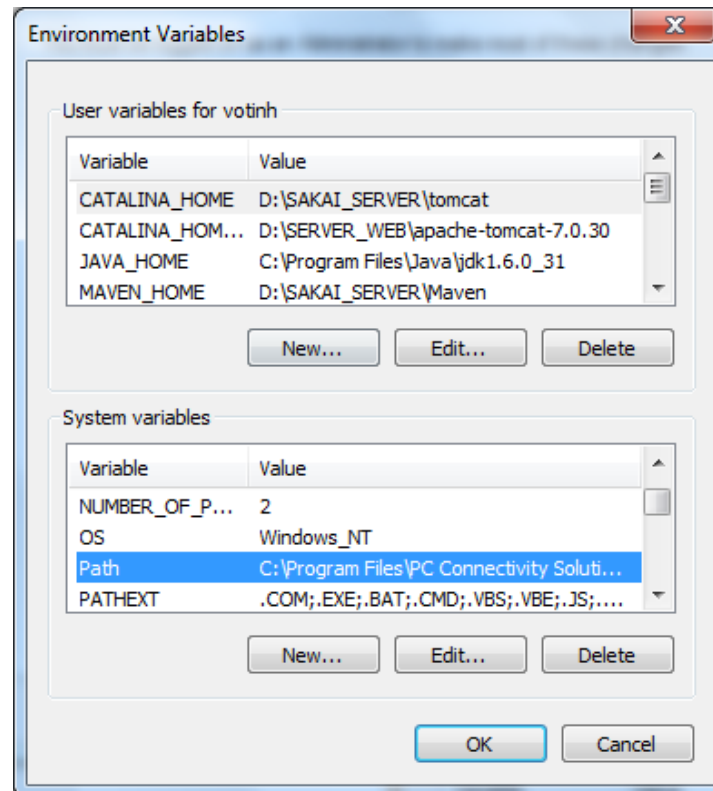
- Trong khung giá trị Variable value, chúng ta copy **đường dẫn cài đặt JDK** vào đây

Sau đó chọn OK, trong vùng **System Variables**, biến **JAVA_HOME** và giá trị của nó sẽ xuất hiện như hình sau:

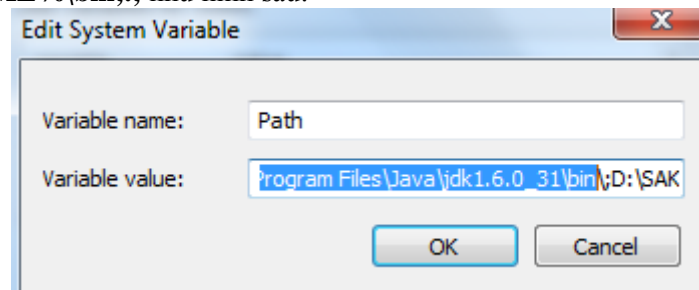


- Bước 5: **Thêm giá trị** với biến **path** trong mục **System Variables** chọn Edit

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows



- Trong mục Variable value, di chuyển tới cuối và nhập vào các giá trị: **%JAVA_HOME%\bin;.**; như hình sau:



- Bước 6: Kiểm tra lại cấu hình

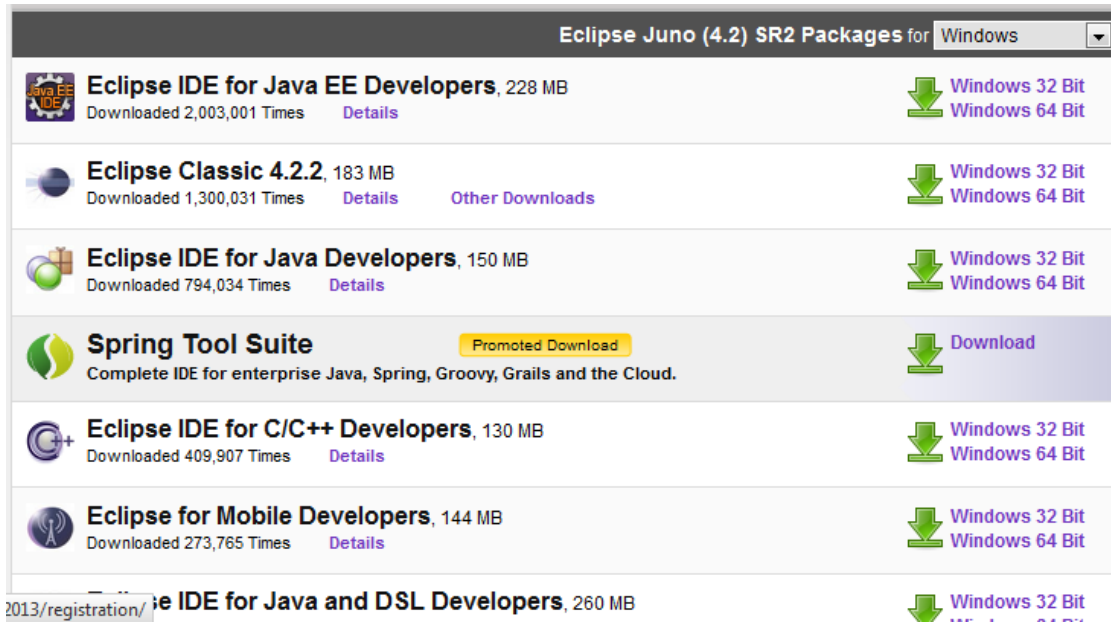
Để kiểm tra version của JDK cài đặt trong máy tính ta vào cmd gõ lệnh: **java -version**

Để kiểm tra javac version ta gõ lệnh: **javac -version**

3. Cài đặt IDE (Eclipse)

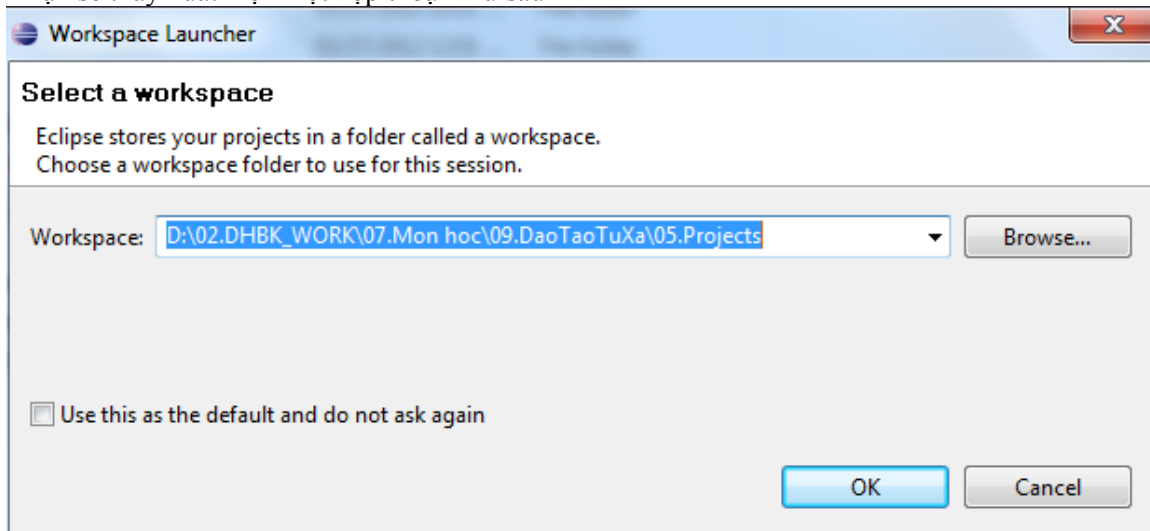
- Bước 1: Download eclipse tại <http://www.eclipse.org/downloads/>

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows



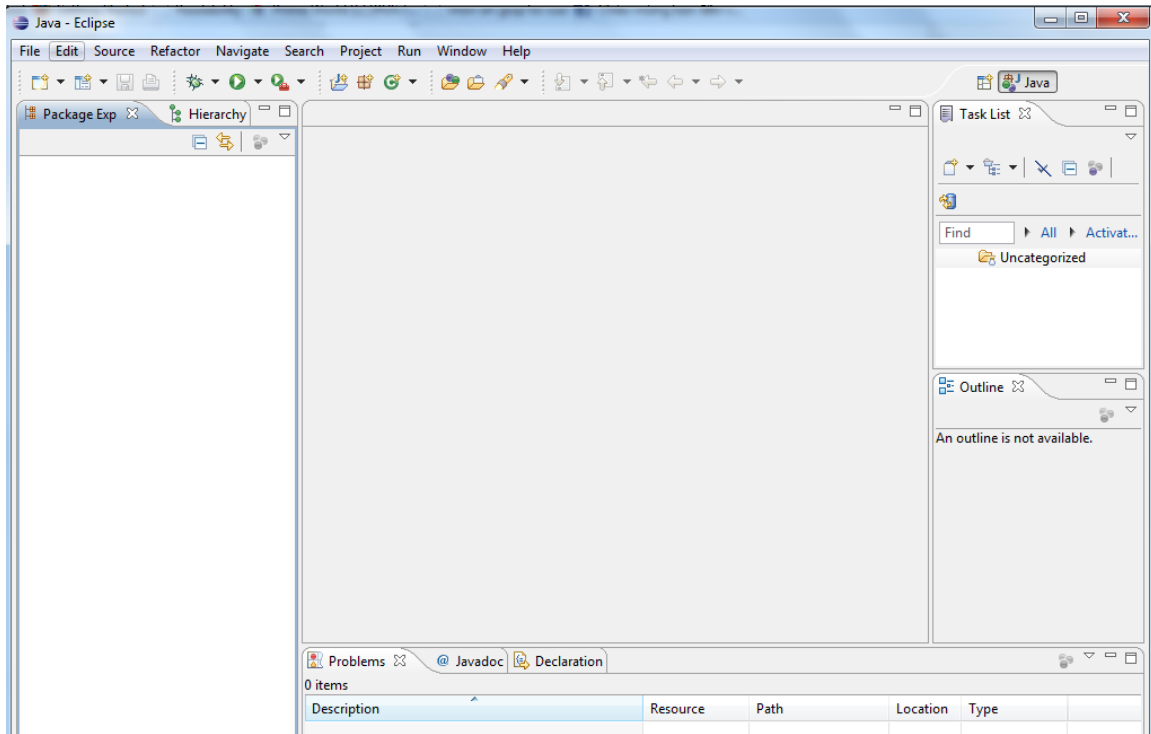
- Bước 2: giải nén file vừa down về ta được thư mục chứa Eclipse
- Bước 3: Bây giờ phần cài đặt đã tiến hành xong bạn hãy vào thư mục vừa giải nén click vào file eclipse.exe

- Bạn sẽ thấy xuất hiện một hộp thoại như sau



- Đây là hộp thoại thông báo cho bạn địa chỉ lưu các bài làm, bạn cũng có thể thay đổi đường dẫn lưu bằng cách bấm chuột vào nút Browse..., khi chọn đường dẫn xong thì bạn bấm OK. Màn hình xuất hiện cửa sổ như sau:

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows

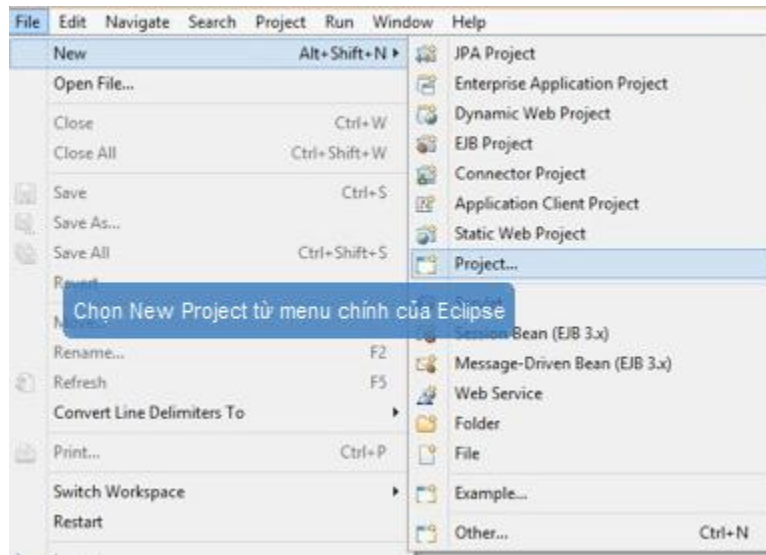


Đó là màn hình giao diện làm việc của Eclipse.

Như thế là chúng ta đã **cài đặt thành công JDK và IDE Eclipse** để chuẩn bị cho **lập trình Java**

4. Tạo một project bằng Eclipse

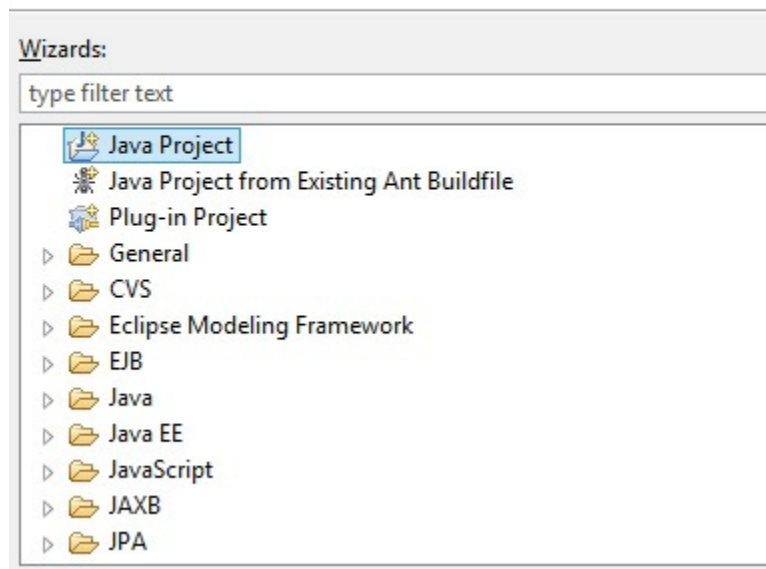
Bước 1: Khởi động Eclipse để vào màn hình làm việc chính của Eclipse như ở trên đã trình bày



Eclipse sẽ hiển thị cửa sổ tùy chọn New Project sau đây:

Select a wizard

Create a Java project



Chọn **Java Project** rồi nhấn **Next** (hoặc double click vào **Java Project**) sẽ xuất hiện cửa sổ **New Java Project** như sau:

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows

Create a Java Project

Create a Java project in the workspace or in an external location.



Project name:

Contents

☒ Create new project in workspace

☐ Create project from existing source

Directory:

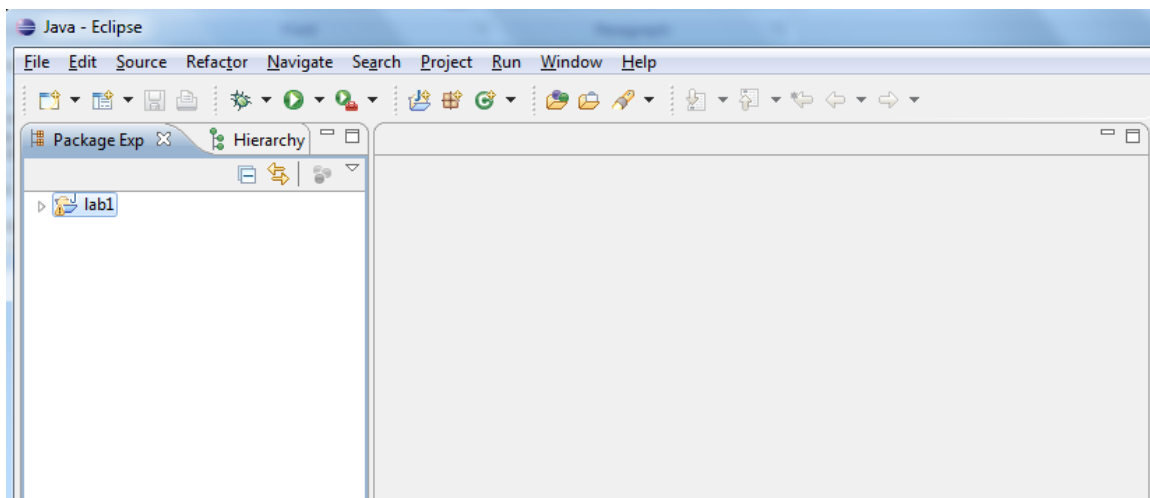
JRE

☐ Use an execution environment JRE:

☒ Use a project specific JRE:

☐ Use default JRE (currently 'jre7') [Configure JREs...](#)

Tại mục **Project name**: nhập tên project cần tạo vào và nhấn nút **Finish**, Eclipse sẽ đưa ta đến màn hình làm việc cùng với project vừa tạo



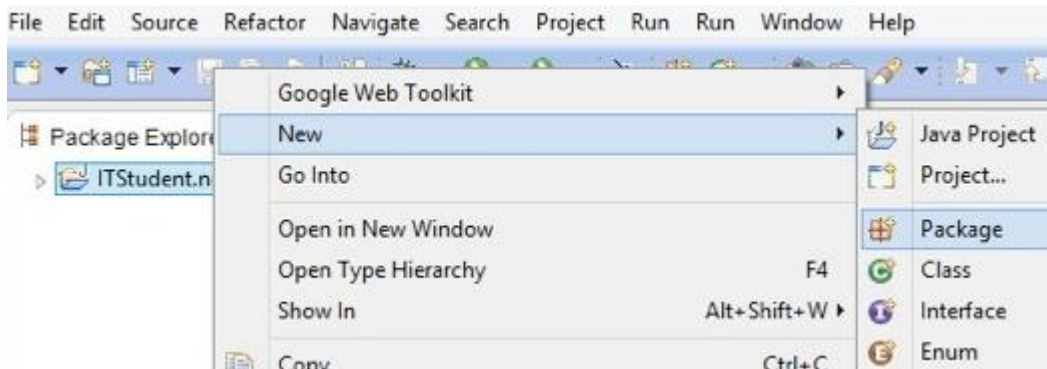
5. Tạo một chương trình java cơ bản

Bước 1: Tạo gói (Java Package)

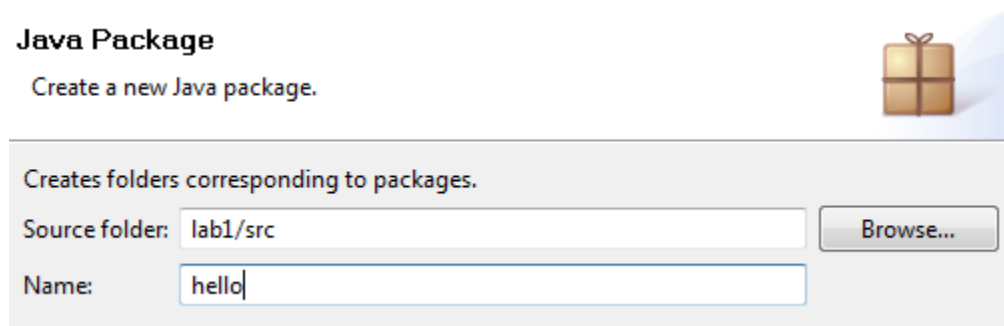
Gói trong lập trình Java là một **nhóm các lớp và giao tiếp** có quan hệ với nhau **được tổ chức như một đơn vị** không gian tên.

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows

Gói giúp phân hoạch không gian tên lớp, giao diện thành những vùng dễ quản lý hơn. Để tạo mới 1 gói trong Eclipse, chúng ta cần click phải vào **project** rồi chọn **New Package**



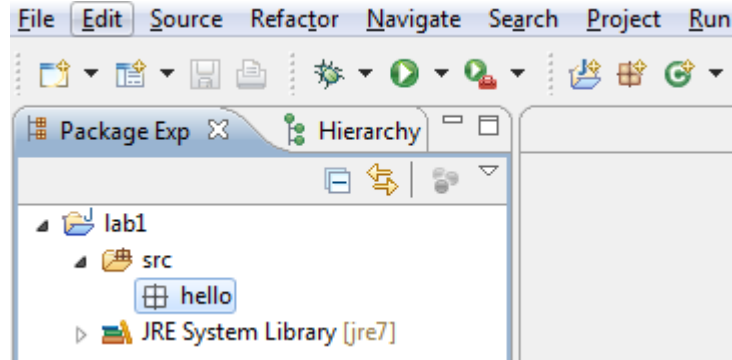
Eclipse sẽ yêu cầu chúng ta chỉ định các thông số cần thiết cho 1 package cần tạo qua cửa sổ sau:



Tại mục Name: nhập tên của gói
Tên gói có một số quy định sau:

- Tên gói phải là chữ thường
- Tên gói không được bắt đầu bằng java hoặc javax
- Không được bắt đầu bằng số hoặc dấu nối (-)

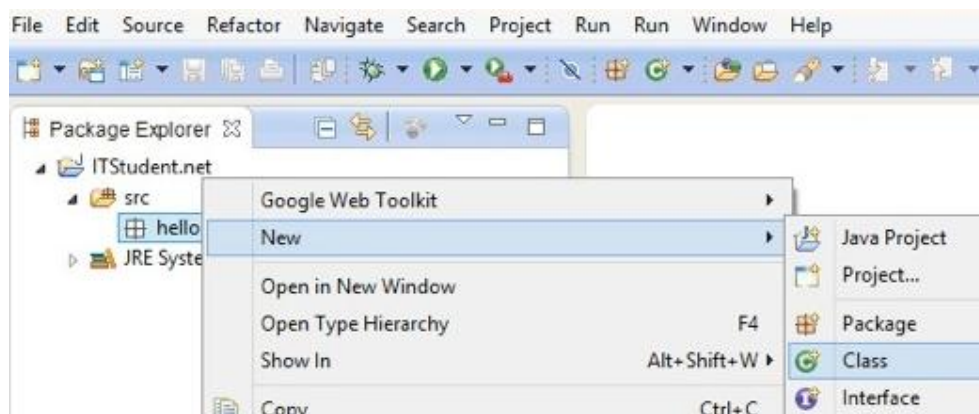
Sau khi nhập tên gói xong thì chọn Finish ta được:



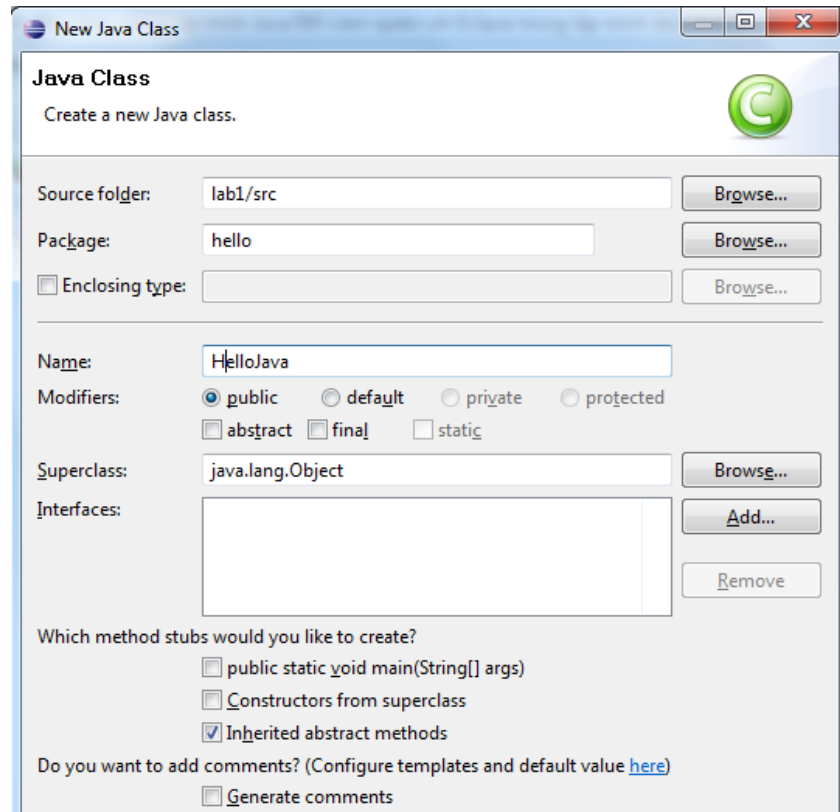
Bước 2: Tạo lớp (**Java Class**)

Trong lập trình Java, lớp (java class) là nơi chứa một tập hợp các đối tượng, constructor, phương thức, hàm, mà ta thực thi lên đó.

Để tạo lớp, chúng ta sẽ click phải vào tên gói và chọn **New Class**



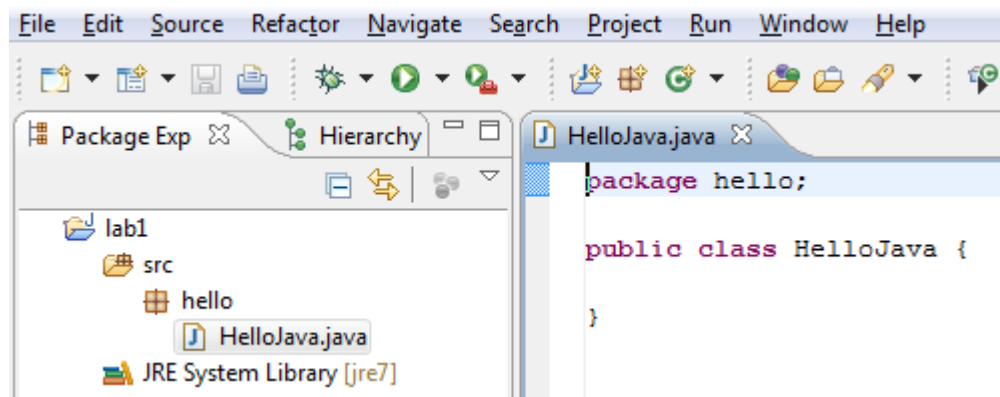
Cửa sổ sau sẽ xuất hiện yêu cầu nhập tên Class vào mục Name:



Tên class có một số quy định như sau:

- Viết hoa chữ cái đầu tiên của mỗi từ
- Không bắt đầu bằng một số, phải bắt đầu bằng ký tự hoặc \$, _
- Không chứa dấu cách hoặc các ký tự toán học: *, /, %,
- Không trùng với tên khác trong cùng gói (tên lớp phân biệt chữ hoa và chữ thường).

Sau khi đặt tên lớp xong chọn **Finish** ta sẽ có:



Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows

Bước 3: Tạo chương trình cơ bản

Để chạy được chương trình thì ta cần phải có hàm `main()` như sau:

```
public class HelloJava {  
    public static void main(String[] args) {  
    }  
}
```

Để in ra kết quả thì ta cần hàm `println()` như sau:

```
public static void main(String[] args) {  
    System.out.println();  
}
```

Ví dụ: nếu muốn in ra chữ “ĐTTX” ta viết như sau:

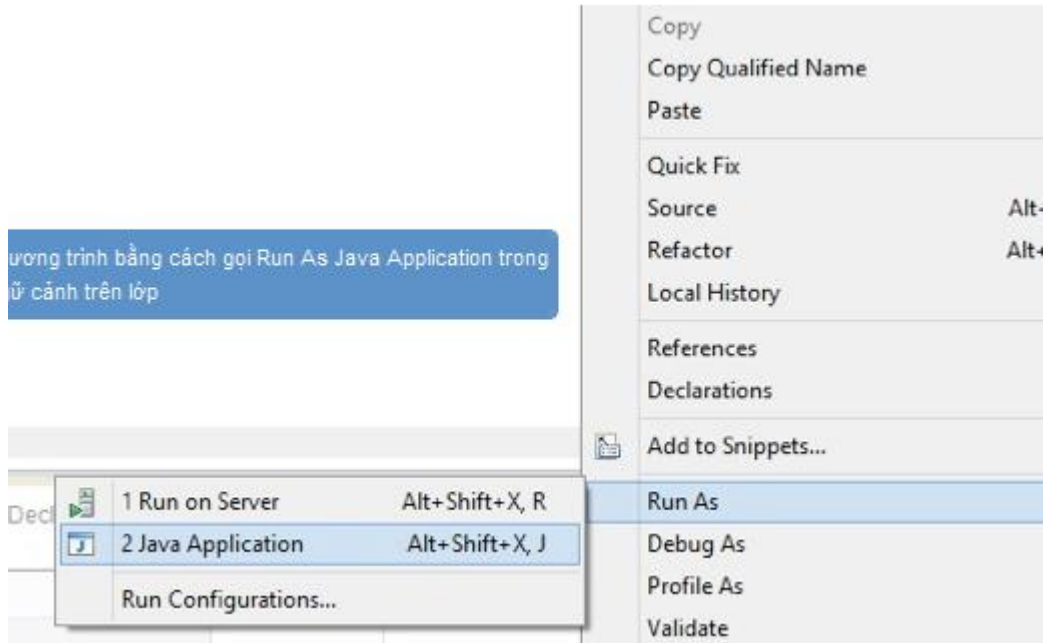
```
public static void main(String[] args) {  
    System.out.println("ĐTTX");  
}
```

Có nhiều cách để có thể chạy một chương trình Java được viết trong Eclipse:

Cách 1: Click vào nút **Run** trên thanh công cụ

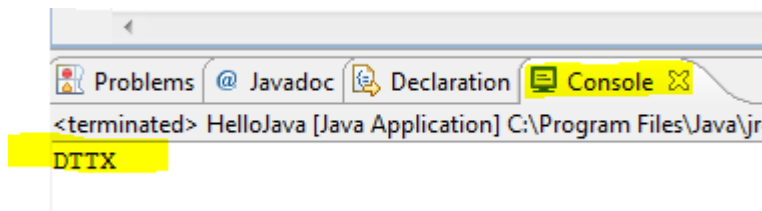


Cách 2: click phải vào lớp chọn **Run As** và **Java Application**



Cách 3: Nhấn tổ hợp phím **Ctrl + F11**

Sau khi chạy kết quả sẽ thu được là:



6. Lập trình socket

6.1. Các kiến thức liên quan đến socket

Socket là gì? Khi nào ta sẽ sử dụng socket?

6.2 Tạo project hiển thị địa chỉ IP của máy tính/ domain

- Tạo project tên “ShowIp”
- Sử dụng packet java.net* để lấy thông tin địa chỉ IP

Code ví dụ lấy Ip của một máy tính:

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows



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

        HostInfo host = new HostInfo();
        host.init();

    }

    public void init() {
        try {
            InetAddress myHost =
                InetAddress.getLocalHost();
            System.out.println(myHost.getHostAddress());
            System.out.println(myHost.getHostName());

        }

        catch (UnknownHostException ex) {
            System.err.println("Cannot find local host");

        }

    }
}
```

Code ví dụ lấy Ip của domain: www.hcmut.edu.vn

```
import java.net.*;
class Info {
    public static void main (String args[]) {
        try {
            InetAddress[] addresses =
                InetAddress.getAllByName("www.hcmut.edu.vn");
            for (int i = 0; i < addresses.length; i++)
            {
                System.out.println(addresses[i]);
            }

        }

        catch (UnknownHostException e) {
            System.out.println("Could not find www.hcmut.edu.vn");

        }

    }
}
```

6.3 Tạo một ứng dụng sever socket

1. Tạo server.

Trong Java ta sử dụng ServerSocket class để tạo một server lắng nghe trên mạng. Khi sử dụng server ta cần có tham số truyền vào là port của ứng dụng (port này do ta tự đặt, thông thường port này có 4 chữ số)

Code ví dụ cho việc tạo server

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows

```
import java.net.*; import
java.io.*; import java.util.Date;
public class DayTimeServer {
    public final static int daytimePort = 5000;
    public static void main(String[] args) {
        ServerSocket theServer;
        Socket theConnection; PrintStream p;
        try {
            theServer = new ServerSocket(daytimePort);
            while (true) {
                theConnection = theServer.accept();
                p = new PrintStream(theConnection.getOutputStream());
                p.println(new Date());
                theConnection.close();
                theServer.close();
            }
        }
        catch (IOException e) {
            System.err.println(e);
        }
    }
}
```

2. Tạo client

Trong Java ta sử dụng socket để kết nối tới server. Để kết nối được với server ta cần phải biết địa chỉ IP của server và port đang mở của ứng dụng.

Code client ví dụ kết nối với server:

```
import java.net.*;
import java.io.*;
public class getSocketInfo {
    public static void main(String[] args) {
        try {
            Socket theSocket = new Socket("localhost", 5000);
            System.out.println("Connected to "
                + theSocket.getInetAddress() + " on port "
                + theSocket.getPort() + " from port "
                + theSocket.getLocalPort() + " of "
                + theSocket.getLocalAddress());
        }

        catch (UnknownHostException e) {
        }
        catch (SocketException e) {
        }
        catch (IOException e) {
        }
    } // end main
} // end getSocketInfo
```

7. Đa luồng trong Java

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows



7.1. Các kiến thức liên quan đến Thread

Thread là gì? Thread được dùng khi nào?

7.2 Tạo một ứng dụng dạng luồng (thread)

Trong Java có nhiều cách tạo luồng trong. Đoạn code sau đây chỉ là một trong những ví dụ tạo luồng (thread) trong máy tính.

Ví dụ: In ra tên của thread khi thread được khởi động. Ta làm như sau:

- Tạo một class có tên MyRun hiện thực của Runnable. Đối tượng của class sẽ được đưa vào thread để chạy.

```
public class MyRun implements Runnable{  
  
    private String name = "my run";  
    /**  
     * phương thức khởi tạo  
     */  
    public MyRun(){  
  
    }  
    /**  
     * phương thức này được gọi khi thread gọi phương thức bắt đầu  
     */  
    public void run (){  
        System.out.println(name);  
    }  
}
```

- Tạo một thread và đưa đối tượng của class MyRun vào Thread.

```
MyRun myrun = new MyRun();  
Thread thread = new Thread(myrun);  
  
thread.start();
```

7.3. Đa luồng trong Java

Để tạo được ứng dụng đa luồng trong Java ta sử dụng nhiều Thread.

Ví dụ: tạo đa luồng in số chẵn và số lẻ trong Java

- Tạo class Inso hiện thực Runnable

Bài thực hành số 03 – Làm quen với môi trường lập trình mạng sử dụng ngôn ngữ lập trình Java trên Windows



```
public class Inso implements Runnable {  
  
    private boolean flag = true;  
    public Inso(boolean flag){  
        this.flag = flag;  
    }  
    @Override  
    public void run() {  
        for(int i = 0; i < 100; i++){  
            if(flag)  
                System.out.println("so chan");  
            else  
                System.out.println("so le");  
  
            try {  
                Thread.sleep(100);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

- Sử dụng Inso class để tạo ứng dụng đa luồng.

```
Inso insochan = new Inso(true);  
Thread thread_chan = new Thread(insochan);  
thread_chan.start();  
  
Inso insole = new Inso(false);  
Thread thread_le = new Thread(insole);  
  
thread_le.start();
```

8. Bài tập:

1. Viết chương trình hiển thị thông tin IP của máy tính
2. Viết chương trình nhận và gửi dữ liệu sử dụng Socket.