



Hackademics Hanoi
Hack your passion, Code your life!

Khóa đào tạo Lập trình Java

Dàn ý

- Giới thiệu Java
- Cài đặt JDK, Eclipse
- Kiểu dữ liệu

1. Giới thiệu Java

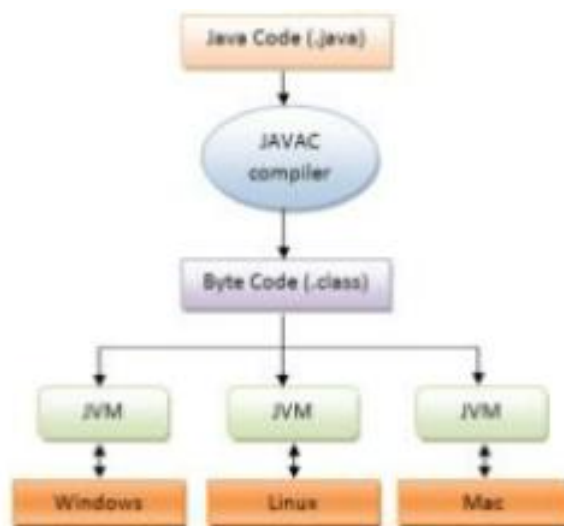
- Chương trình là tập hợp các chỉ thị lệnh yêu cầu máy tính thực hiện tác vụ cụ thể
- Ngôn ngữ lập trình
 - Là một ngôn ngữ lập trình nhân tạo
 - Gồm 1 tập kí hiệu và cú pháp được chuẩn hóa để mô tả những xử lí mà người và máy có thể hiểu được

Giới thiệu Java (Programming)

- Là quá trình xây dựng các chương trình nguồn được viết bằng 1 hoặc nhiều ngôn ngữ lập trình
- Các phương pháp lập trình thường gặp
 - Lập trình tuyến tính (Linear Programming)
 - Lập trình cấu trúc (Structured Programming)
 - Lập trình hướng đối tượng (Object oriented programming)
 - Lập trình trực quan (Visual programming)

Giới thiệu ứng dụng Java

- Ngôn ngữ trung gian của Java – Java bytecode
 - Là định dạng biên dịch chương trình viết bằng Java
 - Là các chỉ lệnh JVM thực thi
 - Các tập tin bytecode có phần mở rộng .class



Lịch sử phát triển của Java

- Lịch sử phát triển của Java
 - Được viết bởi James Gosling vào 6/1991
 - Ban đầu gọi là Oak
 - Được đổi tên là Java 1.0 năm 1995
- “Write once, Run everywhere”
- Phiên bản mới nhất là Java SE 7, Java EE 7

2. Môi trường phát triển

- JDK: Là công cụ để phát triển Java, bao gồm
 - JRE: Môi trường chạy ứng dụng Java
 - Javac: Chương trình để biên dịch mã thành bytecode
 - Archive: Là một chương trình nén các file thành một file duy nhất có đuôi jar. Thường dùng để đóng gói các file class
 - JavaDoc: Công cụ tạo ra tài liệu hướng dẫn sử dụng API
- Giới thiệu Eclipse
 - Là môi trường phát triển tích hợp (IDE – Integrated Development Environment)
 - Được phát triển bởi IBM, và hiện nay là tổ chức Eclipse
 - Chứa workspace và hệ thống plugins mở rộng

Cài đặt JDK

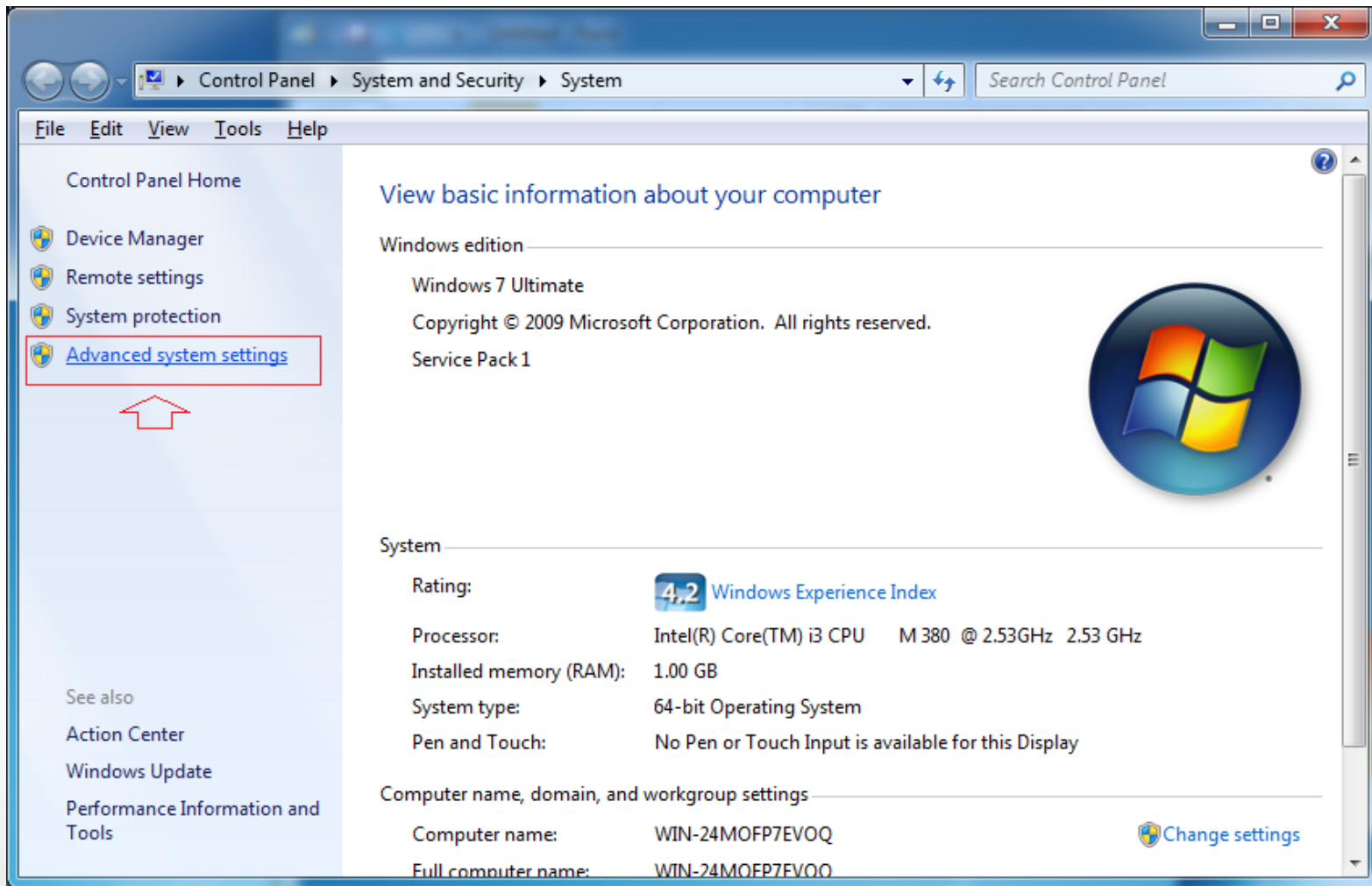
- Vào trang java download

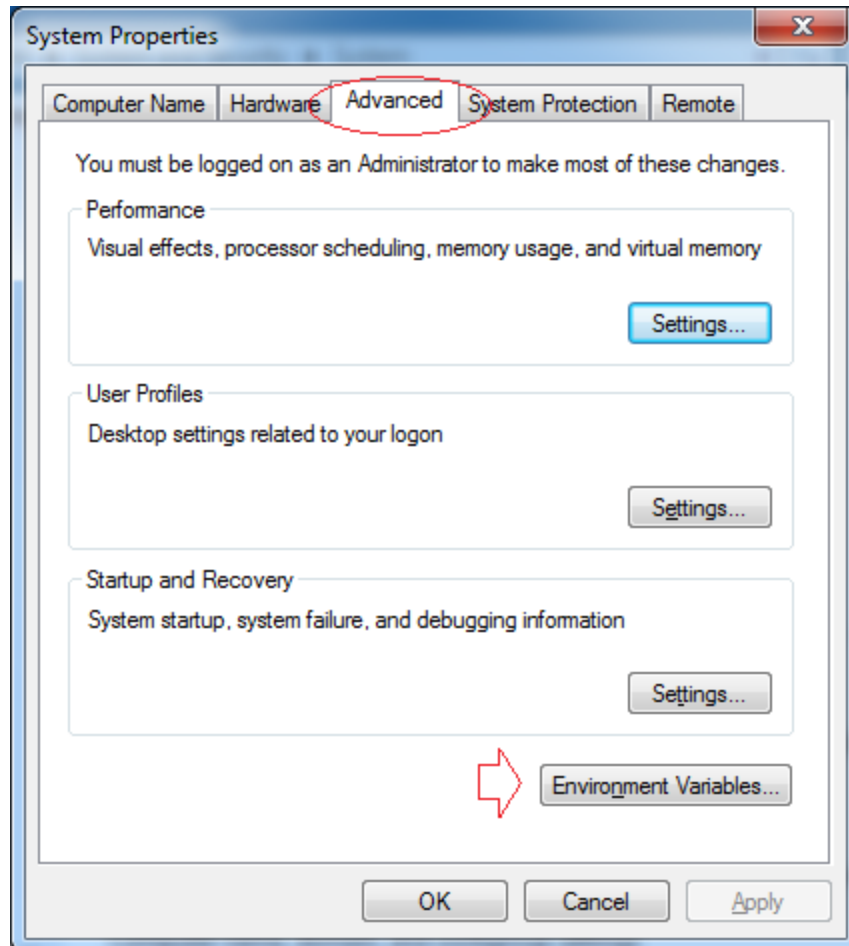
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

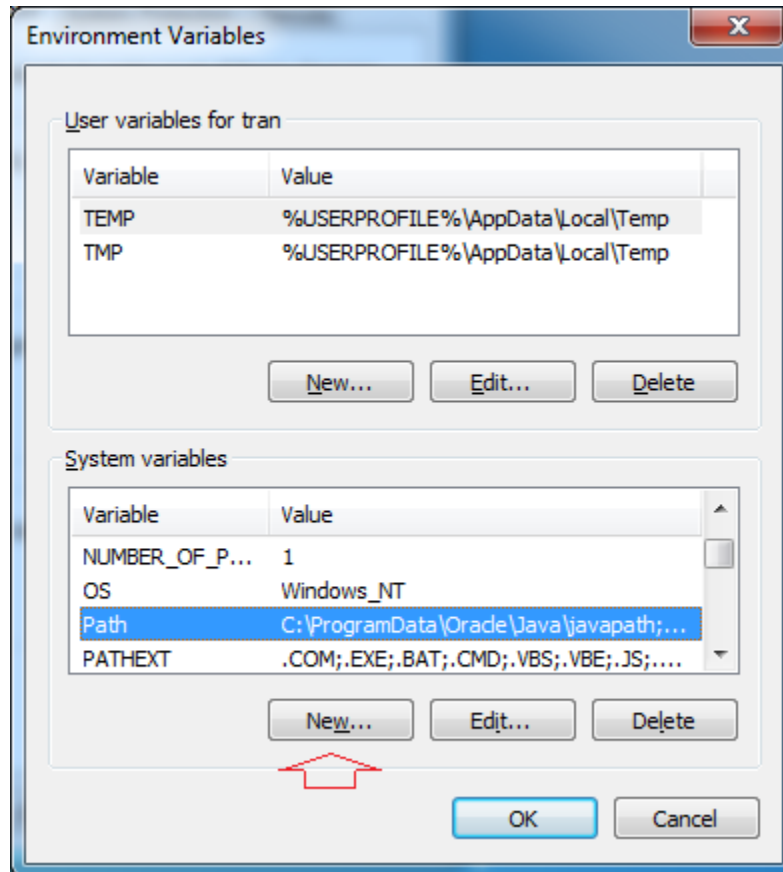
Java SE Development Kit 8u101		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

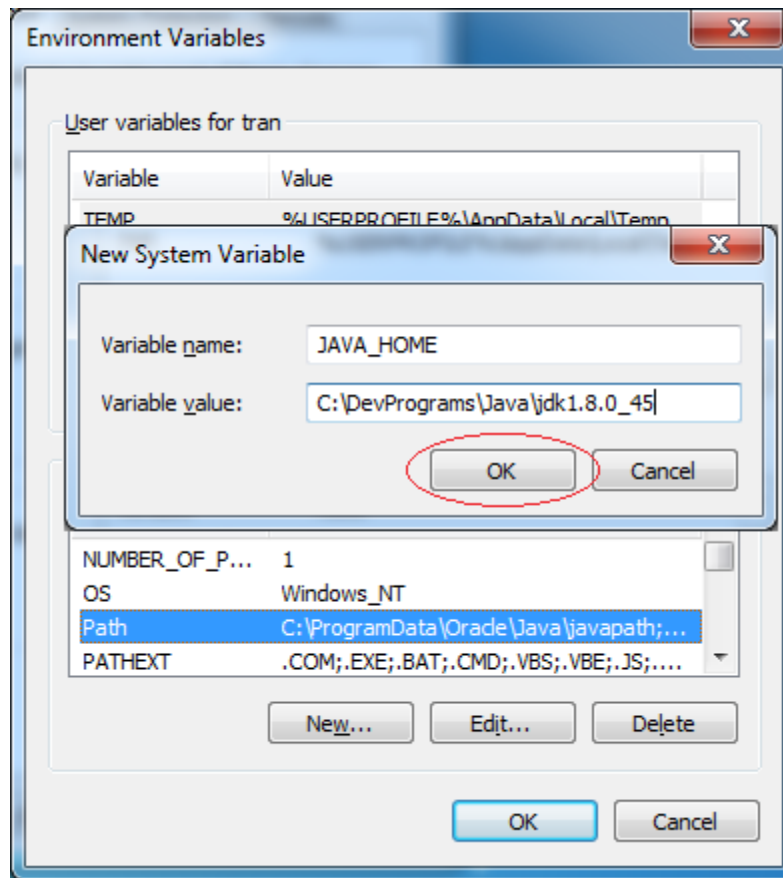
- Chọn accept và download bản ứng với hệ điều hành

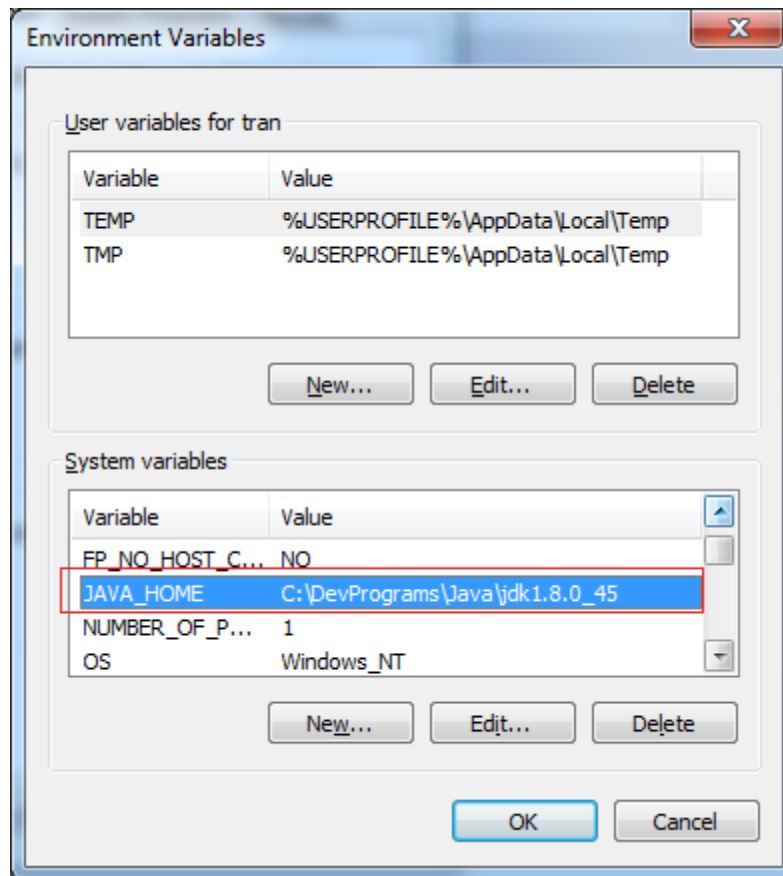
Chỉnh environment var.

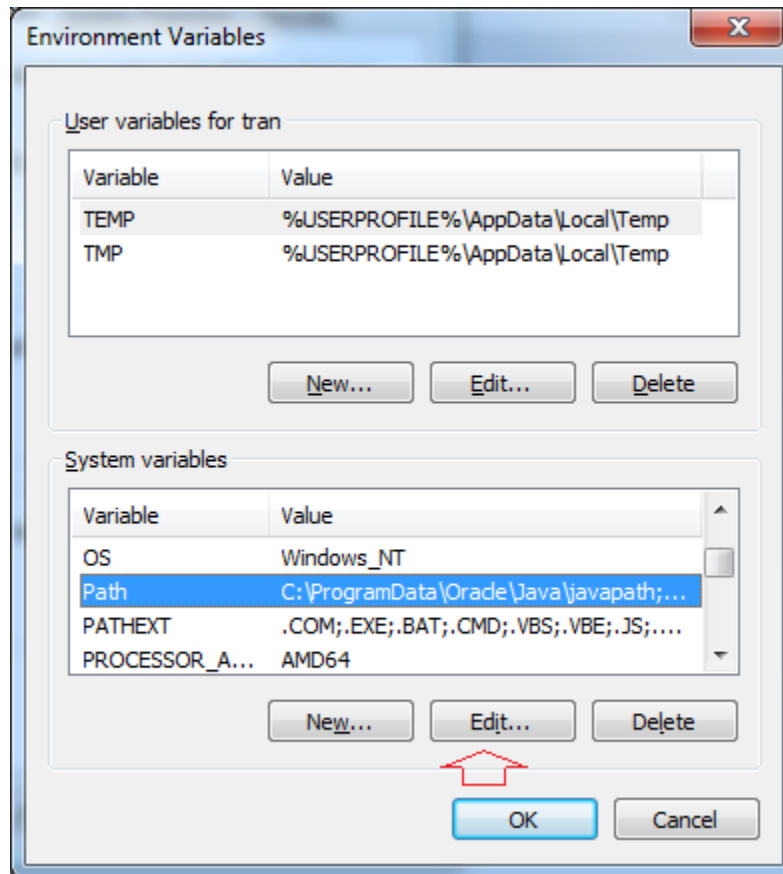


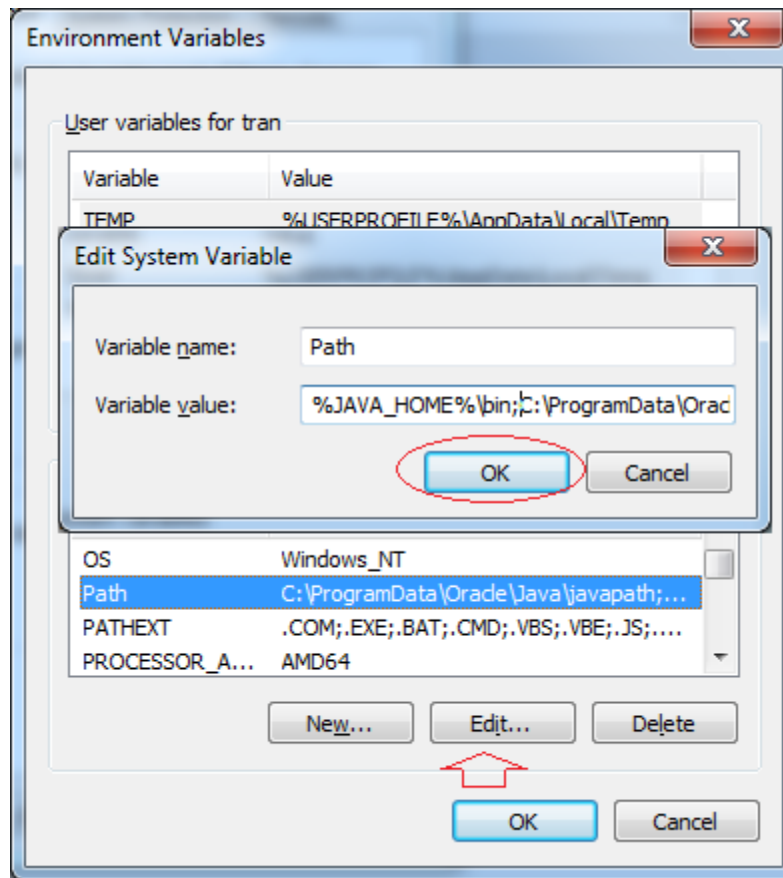


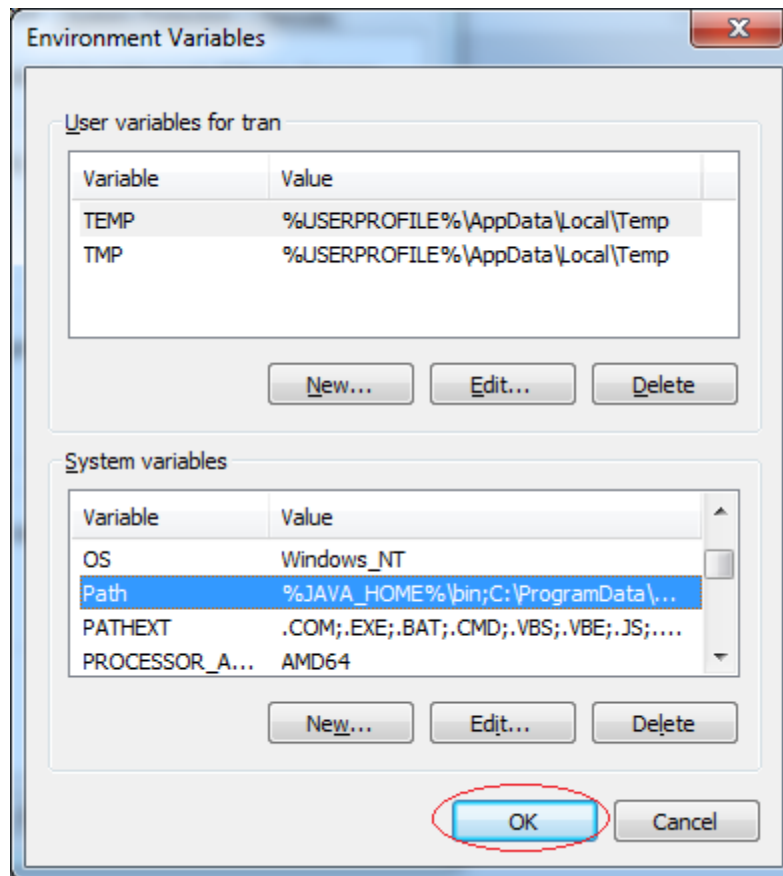












Kiểm tra

```
cmd C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\camnh>java
Usage: java [-options] class [args...]
           (to execute a class)
 or  java [-options] -jar jarfile [args...]
           (to execute a jar file)
where options include:
    -d32          use a 32-bit data model if available
    -d64          use a 64-bit data model if available
    -server       to select the "server" VM
                  The default VM is server.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  A ; separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -D<name>=<value>
                  set a system property
    -verbose:[class|gc|jni]
                  enable verbose output
    -version      print product version and exit
    -version:<value>
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  require the specified version to run
    -showversion  print product version and continue
    -jre-restrict-search | -no-jre-restrict-search
                  Warning: this feature is deprecated and will be removed
```

Cài đặt Eclipse

- Vào trang chủ <https://eclipse.org/downloads/>

□ DOWNLOAD

Download from: Taiwan - Computer Center, Shu-Te University (http)

File: eclipse-inst-win64.exe SHA-512

>> Select Another Mirror

Chạy setup.exe

type filter text



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration



Eclipse IDE for Java EE Developers

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others.



Eclipse IDE for C/C++ Developers

An IDE for C/C++ developers with Mylyn integration.



Eclipse IDE for JavaScript and Web Developers

The essential tools for any JavaScript developer, including JavaScript language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

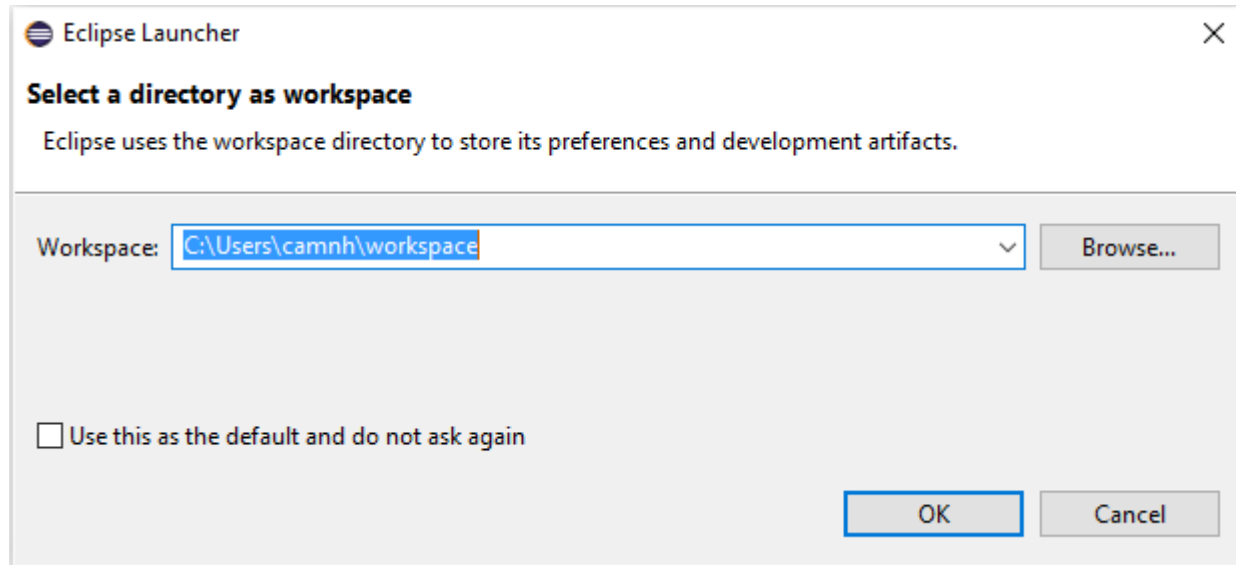


Eclipse IDE for PHP Developers

The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

Cài đặt eclipse

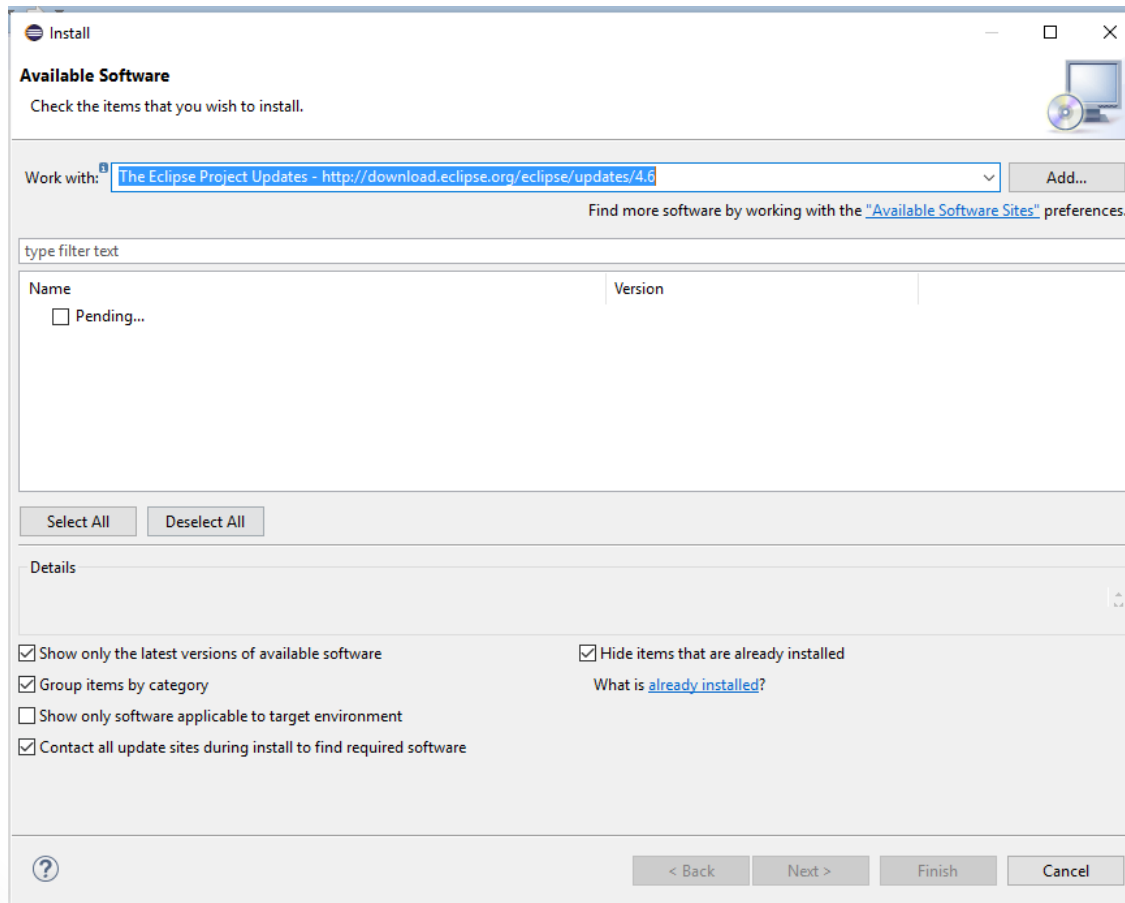
- Thiết lập workspace



Cài đặt plugins

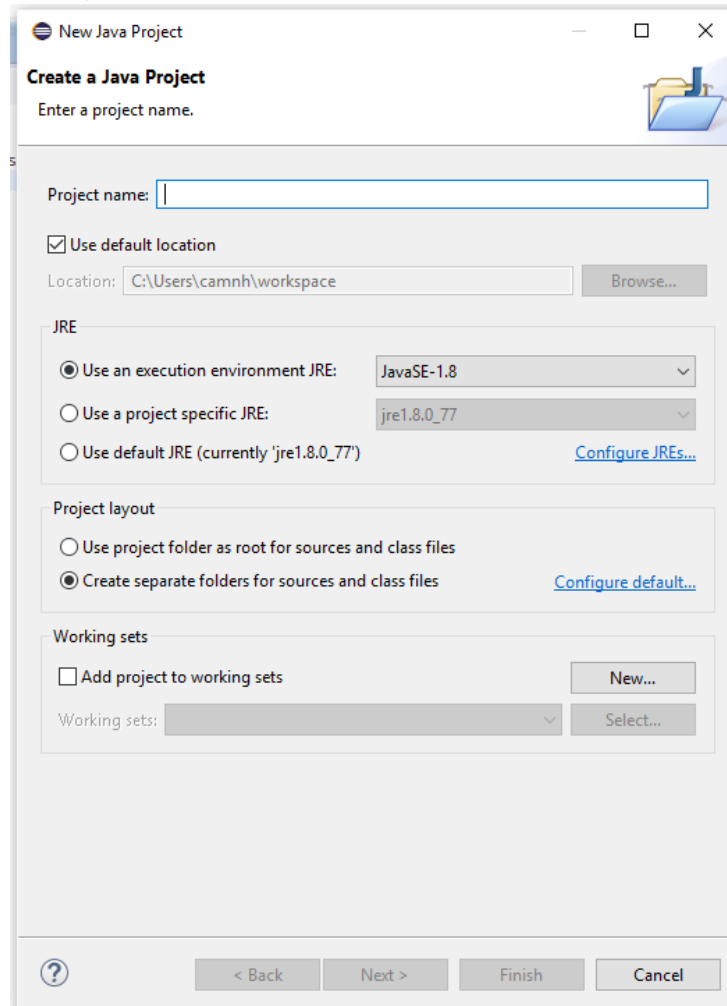
- Thành phần được gắn với Eclipse để thực hiện, hỗ trợ một công việc nào đó.

Help > Install new software.



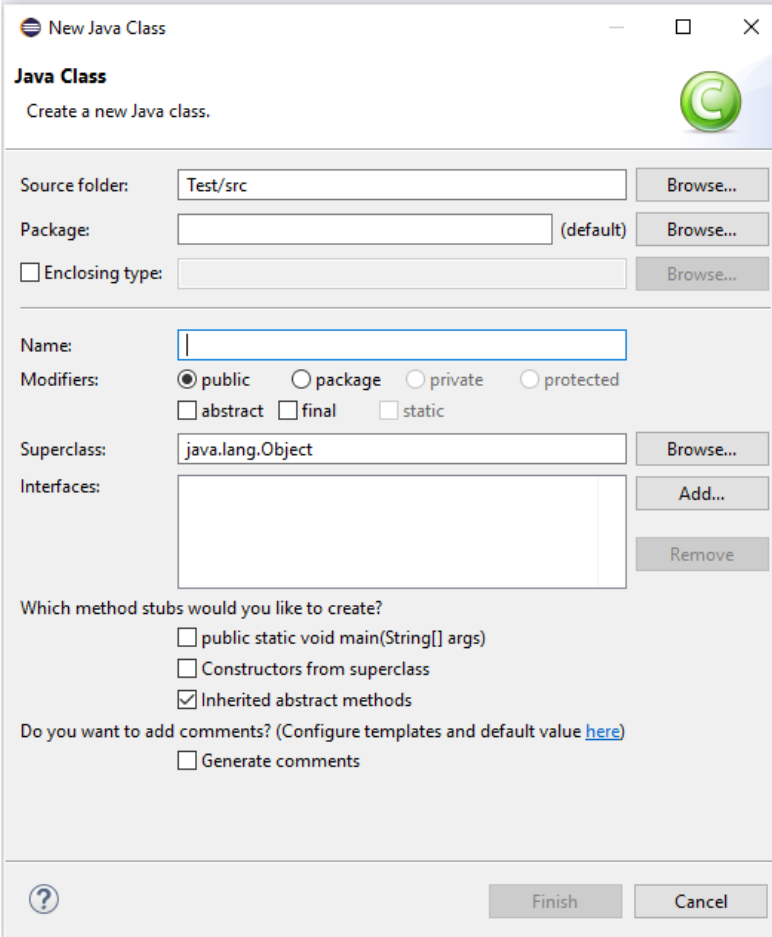
Xây dựng ứng dụng đầu tiên

- File > new Java Project



Tạo class mới

- Chuột phải vào src > new Class



The screenshot shows the 'New Java Class' dialog box. At the top, it says 'Java Class' and 'Create a new Java class.' with a green 'C' icon. The 'Source folder' is 'Test/src' with a 'Browse...' button. The 'Package' is '(default)' with a 'Browse...' button. There is an unchecked checkbox for 'Enclosing type' with a 'Browse...' button. The 'Name' field is empty. The 'Modifiers' section has radio buttons for 'public' (selected), 'package', 'private', and 'protected', and checkboxes for 'abstract', 'final', and 'static'. The 'Superclass' is 'java.lang.Object' with a 'Browse...' button. The 'Interfaces' section has an 'Add...' button and a 'Remove' button. Below this, it asks 'Which method stubs would you like to create?' with checkboxes for 'public static void main(String[] args)', 'Constructors from superclass', and 'Inherited abstract methods' (checked). It then asks 'Do you want to add comments? (Configure templates and default value [here](#))' with a checkbox for 'Generate comments'. At the bottom are 'Finish' and 'Cancel' buttons.

New Java Class

Java Class
Create a new Java class.

Source folder: Test/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

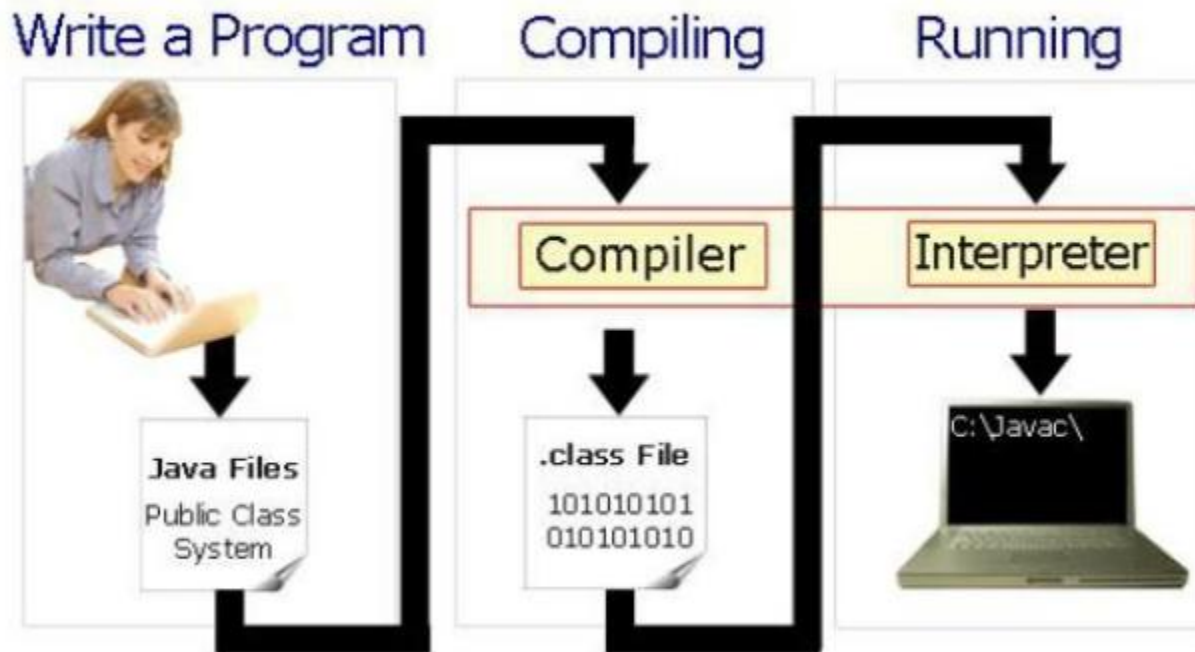
Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Finish Cancel

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

- Class Run: Tên lớp
- Main: Tên phương thức chính, điểm bắt đầu của chương trình chạy bằng Java.
- `System.out.println("Hello")` : In dòng chữ "Hello" ra màn hình console.

Xây dựng và thực thi



Bài tập

- Cài đặt JDK, eclipse
- Viết ứng dụng đầu tiên ghi chuỗi Hello World ra màn hình

- Trang này để trống

Kiểu dữ liệu

- Kiểu dữ liệu và chuyển đổi kiểu dữ liệu
- Biến và hằng số
- Các xử lý trên kiểu String.

Kiểu dữ liệu cơ sở

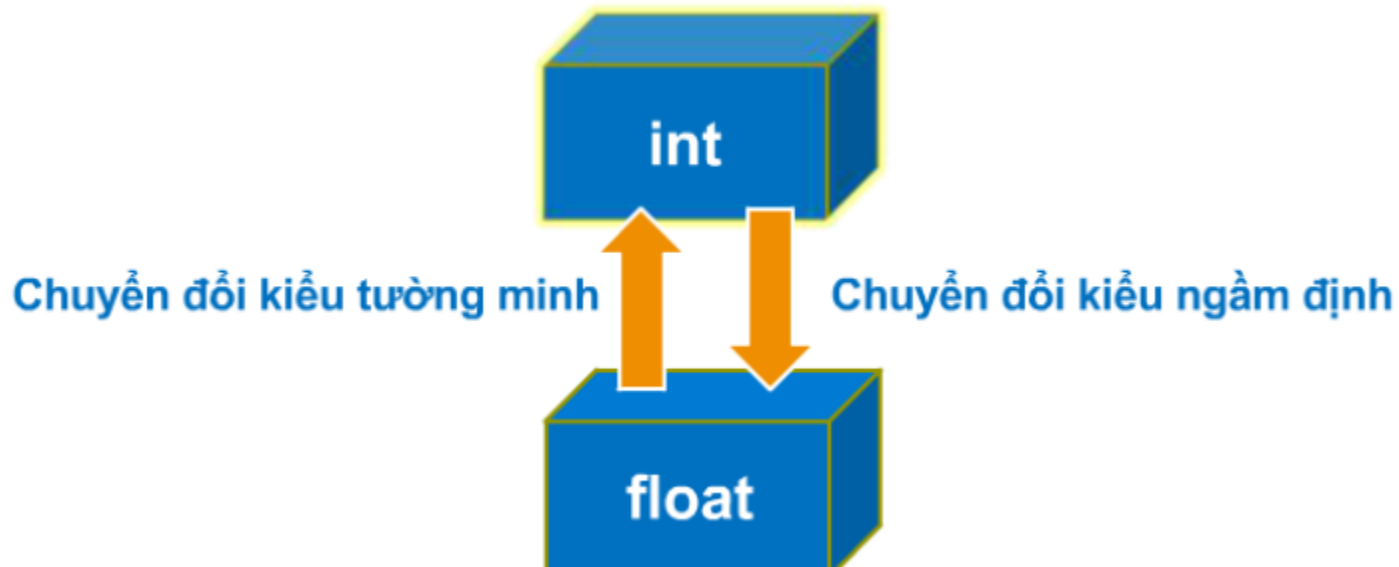
Kiểu dữ liệu	Kích thước	Miền giá trị
Byte	8 bit	$-2^7..2^7-1$
Short	16 bit	$-2^{15}..2^{15}-1$
Int	32 bit	$-2^{31}..2^{31}-1$
Long	64 bit	$-2^{63}..2^{63}-1$
Float	32 bit	$3.4e-038 .. 3.4e+038$
Double	64 bit	$1.7e-308 .. 1.7e+308$
Boolean	1 bit	True/false
Char	16 bit	Kí tự unicode 16 bit

Kiểu dữ liệu tham chiếu

Kiểu dữ liệu	Mô tả
Array	Kiểu mảng
Class	Lớp đối tượng
Interface	Giao diện
String	Chuỗi kí tự

Chuyển đổi kiểu dữ liệu

- Cơ chế chuyển đổi kiểu dữ liệu



Cơ chế chuyển đổi

- Chuyển đổi kiểu ngầm định
 - Hai kiểu tương thích
 - Kiểu đích có tầm giá trị lớn hơn kiểu nguồn, trình biên dịch tự động chuyển kiểu

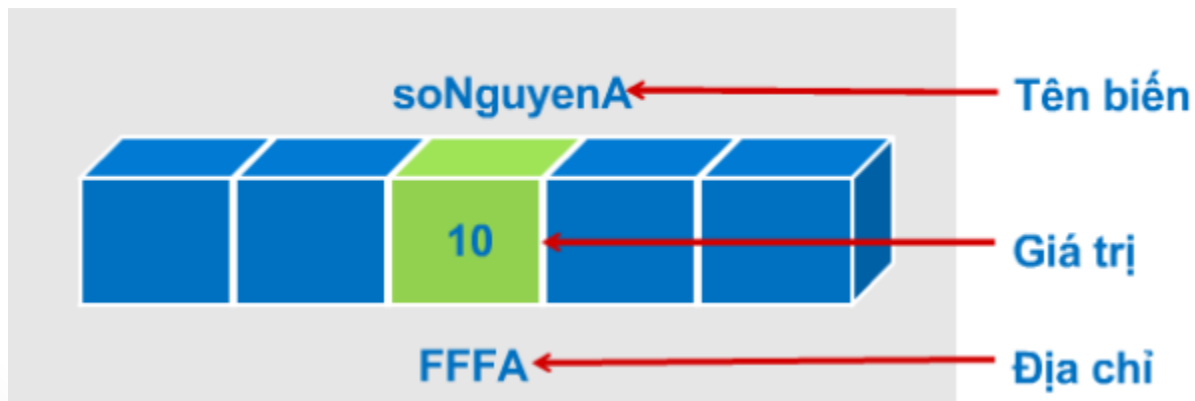
```
int i = 1000;  
long p = i;
```

- Chuyển đổi kiểu tường minh
 - Chuyển đổi từ kiểu dữ liệu có độ chính xác cao sang kiểu có độ chính xác thấp hơn, người dung tự tiến hành chuyển kiểu

```
long l = 1000;  
int i = (int) l;
```


Biến và hằng số

- Khái niệm biến
 - Là một đơn vị lưu trữ trên bộ nhớ của máy tính, lưu trữ các giá trị có thể được dùng để tính toán xử lý
 - Biến có thể lưu trữ dữ liệu dạng chuỗi, dạng số, dạng ngày giờ.. Tùy thuộc vào kiểu dữ liệu
 - Biến được khai báo trước khi dùng



Khai báo biến và hằng số

- Cú pháp: kiểu_dữ_liệu = tên_biến

- Ví dụ:

```
int tuoi;  
String ten;  
double luong;
```

- **Hằng số:** là giá trị không thay đổi trong suốt quá trình chạy ứng dụng
- **Cú pháp: final KieuDuLieu TenHang;**
- **Ví dụ:**

```
final Double PI = 3.14;
```

Quy tắc đặt tên

- Biến và hằng số:
 - Có thể bao gồm các kí tự chữ, số, kí tự, dấu “_” và dấu “\$”.
 - Bắt đầu bằng kí tự chữ
 - Không trùng với từ khóa trong Java
 - Phân biệt chữ hoa, thường
 - Nếu tên biến gồm 1 từ đơn: viết chữ thường
 - Nếu tên biến gồm nhiều từ: kí tự bắt đầu của từ t2 viết hoa

Xử lý chuỗi (String)

- Khai báo

```
String b = "Hello";
```

- Java có thư viện đi kèm, giúp cho vấn đề xử lý chuỗi trở nên dễ dàng hơn

Phương thức xử lý kiểu String

Vấn đề:

Cần có các giá trị để phục vụ cho việc hiển thị và tính toán

Ví dụ:

- Tính chiều dài của chuỗi s
- Nối chuỗi s1 vào chuỗi s
- Lấy một ký tự tại vị trí index trong chuỗi s
- So sánh hai chuỗi s1 và s2
- Tìm vị trí xuất hiện đầu tiên của chuỗi s2 trong chuỗi s

Giải quyết:

Sử dụng hàm chuỗi trong thư viện hàm của Java

```
String s = "Happy ";
String s1 = "New Year";
// Tính chiều dài chuỗi
int len = s.length(); // 6
// Nối chuỗi s1 vào chuỗi s: tương đương s + s1
s.concat(s1); // Happy New Year
// Lấy một ký tự tại vị trí số 8 của s
char result = s.charAt(8); // e
// So sánh hai chuỗi s1 và s2
String s2 = "New Year";
s2.compareTo(s1); // 0 (trả về 0, <0 hoặc >0)
// Tìm vị trí xuất hiện đầu tiên của chuỗi s2 trong chuỗi s
s.indexOf(s2); // 6 (vị trí đầu tiên)
```

Phương thức xử lý kiểu String

Vấn đề:

Cần có các giá trị để phục vụ cho việc hiển thị và tính toán

Ví dụ:

- Tìm vị trí xuất hiện cuối cùng của chuỗi s2 trong chuỗi s
- Thay thế chuỗi s1 bằng chuỗi s2 trong chuỗi s
- Loại bỏ các khoảng trắng thừa của chuỗi s3
- Tạo chuỗi con s4 từ chuỗi s từ vị trí số 6

Giải quyết:

Sử dụng hàm chuỗi trong thư viện hàm của Java

```
String s = "Happy New Year";
String s1 = "Happy";
String s2 = "New";

// Tìm vị trí xuất hiện cuối cùng
// của chuỗi s2 trong chuỗi s
s.lastIndexOf(s2); // 6 (trả về -1
// nếu không tìm thấy)

// Thay thế chuỗi s1 bằng chuỗi s2
// trong chuỗi s
s.replace(s1, s2); // New New Year

// Loại bỏ các khoảng trắng thừa của
// chuỗi s3
String s3 = "    Hello Bi    ";
s3.trim(); // Hello Bi

// Tạo chuỗi con s4 từ chuỗi s từ vị
// trí số 6
String s4 = s.substring(6); // New
Year
```

StringBuilder

- Quản lí chuỗi có thể thay đổi kích thước lẫn nội dung

```
// Khởi tạo  
StringBuilder()  
StringBuilder(int capacity)  
StringBuilder(String s)
```

```
// Phương thức  
append()  
insert()  
delete()  
reverse()
```

Ví dụ

```
public class Run {  
  
    public static void main(String[] args){  
        StringBuilder sb = new StringBuilder();  
        sb.append("Welcome ");  
        sb.append("to ");  
        sb.append("java");  
        System.out.println(sb);  
    }  
}
```

Problems @ Javadoc Declaration Console

<terminated> Run [Java Application] C:\Program Files\Java\jre1.8.0_77\bin\j
Welcome to java

Các kiểu xử lí trên String

□ Lớp **StringBuilder**

- **append()**
 - **append(char c)** : gắn thêm chuỗi đại diện của ký tự c vào chuỗi
 - **append(int i)** : gắn thêm chuỗi đại diện của số nguyên i vào chuỗi
 - **append(Object obj)** : gắn thêm chuỗi đại diện của đối tượng obj vào chuỗi
 - **append(String s)** : gắn thêm chuỗi cụ thể s vào chuỗi

Các kiểu xử lý trên String

❑ Lớp StringBuilder

- insert()

- `insert(int offset, char c)`: chèn chuỗi đại diện của ký tự c vào chuỗi
- `insert(int offset int i)`: chèn chuỗi đại diện của số nguyên i vào chuỗi
- `insert(int offset, Object obj)`: chèn chuỗi đại diện của đối tượng obj vào chuỗi
- `append(int offset, String s)`: chèn chuỗi cụ thể s vào chuỗi

❑ Lớp StringBuilder

- delete()
 - `delete(int start, int end)`: xóa các ký tự từ **start** tới **end** ra khỏi chuỗi
 - `deleteCharAt(int index)`: xóa ký tự tại vị trí **index** ra khỏi chuỗi
- `reverse()`: đảo ngược chuỗi trong đối tượng đang có, có kết quả trả về là một tham chiếu đến đối tượng này

StringTokenizer

- Chia chuỗi thành các chuỗi con

```
// Khởi tạo  
StringTokenizer(String str)  
StringTokenizer(String  
str,String delim)  
StringTokenizer(String  
str,String delim,boolean  
returnDelims)
```

```
// Phương thức  
countTokens()  
hasMoreTokens()  
nextToken()  
hasMoreElements()  
nextElement()
```

Xử lý trên StringTokenizer

□ Lớp StringTokenizer

- Các hàm khởi tạo của lớp
 - `StringTokenizer(String str)` : xây dựng một chuỗi tokenizer cho một chuỗi cụ thể str. Sử dụng các **delim** mặc định là: `"\t\n\r\f"`
 - `StringTokenizer(String str, String delim)` : xây dựng một chuỗi tokenizer cho một chuỗi cụ thể str. Các ký tự trong delim là ký tự để phân tách các token.
 - `StringTokenizer(String str, String delim, boolean returnDelims)` : xây dựng một chuỗi tokenizer cho một chuỗi cụ thể str. Nếu `returnDelims = true` thì mỗi delim được trả về là một chuỗi có chiều dài =1, ngược lại thì delim sẽ được bỏ qua và xem như là một dấu phân cách giữa các token

```
import java.util.StringTokenizer;

public class Run {

    public static void main(String[] args){
        String s = "Lập trình Java";
        StringTokenizer sT = new StringTokenizer(s);
        while(sT.hasMoreTokens()){
            System.out.println(sT.nextToken());
        }

        s = "Lập/trình/Java";
        sT = new StringTokenizer(s, "/");
        while(sT.hasMoreTokens()){
            System.out.println(sT.nextToken());
        }
    }
}
```

Problems @ Javadoc Declaration Console

<terminated> Run [Java Application] C:\Program Files\Java\jre1.8.0_77\bin\javaw.exe

Lập
trình
Java
Lập
trình
Java

❑ Lớp StringTokenizer

- `countTokens()`: tính số lượng token trong chuỗi còn lại khi sử dụng delim mặc định
- `hasMoreTokens()`: kiểm tra xem có còn token trong chuỗi các token hay không. (*)
- `nextToken()`: trả về token tiếp theo trong chuỗi các token (**)
- `hasMoreElements()`: tương tự như (*), tuy nhiên nó ở trong `Enumeration<Object>`
- `nextElement()`: tương tự như (**), tuy nhiên giá trị trả về là `Object`

Điều khiển chương trình

- Câu lệnh rẽ nhánh: Nếu...thì...
- Chỉ chạy 1 lần nếu `condition` là đúng, nếu không thì sẽ chạy về `else`

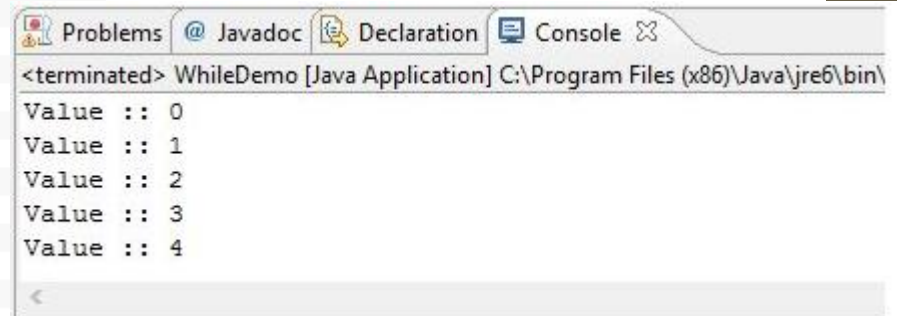
```
if(condition)
    statements;
else if (condition)
    statements;
else if(condition)
    statement;
else
    statements;
```


Điều khiển chương trình

- Câu lệnh lặp: while/do..while/for /foreach
- While: chạy liên tục nếu như condition đúng.

```
While (condition) {  
    //statement  
}
```

```
01. package com.deepak.main;  
02.  
03. public class WhileDemo  
04. {  
05.     public static void main( String[] args )  
06.     {  
07.         int i = 0;  
08.         while ( i < 5 )  
09.         {  
10.             System.out.println( "Value :: " + i );  
11.             i++;  
12.         }  
13.     }  
14. }
```



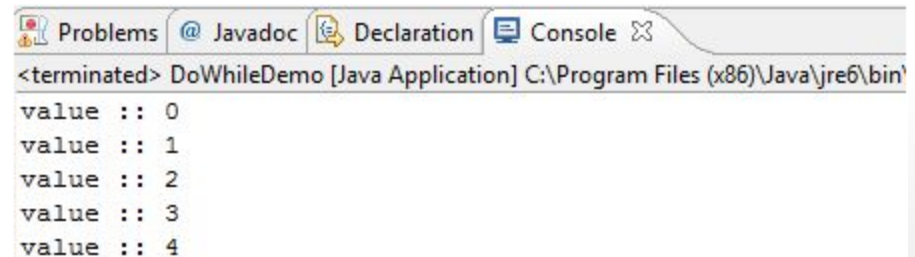
Điều khiển chương trình

- Do..while(): Giống while, nhưng điều kiện kiểm tra ở cuối cùng, đảm bảo vòng lặp chạy ít nhất 1 lần

Example

```
01. package com.deepak.main;
02.
03. public class DoWhileDemo
04. {
05.     public static void main( String[] args )
06.     {
07.         int i = 0;
08.         do
09.         {
10.             System.out.println( "value :: " + i );
11.             i++;
12.         }
13.         while ( i < 5);
14.     }
15. }
```

Output



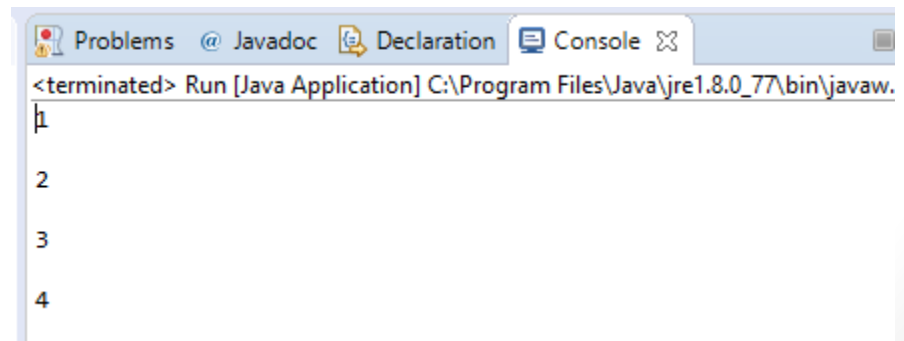
The screenshot shows an IDE window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the Java application. The output consists of five lines, each showing the value of 'i' from 0 to 4, followed by a double colon and the word 'value'. The window title bar indicates the application is 'DoWhileDemo [Java Application]' running in the directory 'C:\Program Files (x86)\Java\jre6\bin'.

```
<terminated> DoWhileDemo [Java Application] C:\Program Files (x86)\Java\jre6\bin'
value :: 0
value :: 1
value :: 2
value :: 3
value :: 4
```

- For: Chạy vòng lặp với điều kiện bắt đầu và điều kiện kết thúc.

```
for(<initialization>;<condition>;<increment or decrement statement>){  
    <block of code>  
}
```

```
public static void main(String[] args){  
    for(int i = 1; i<5;i++){  
        System.out.println(i+"\n");  
    }  
}
```



The screenshot shows an IDE window with a tab labeled 'Console'. The console output displays the numbers 1, 2, 3, and 4, each on a new line, followed by a blank line. The command prompt shows the execution of a Java application using the javaw command.

```
<terminated> Run [Java Application] C:\Program Files\Java\jre1.8.0_77\bin\javaw.  
1  
2  
3  
4
```

Bài tập chuỗi

- Nhập vào một chuỗi, ví dụ “ Hackademics ” hãy loại bỏ những khoảng trắng thừa trong chuỗi.
- Đếm có bao nhiêu khoảng trắng trong chuỗi.
- Viết chương trình đảo ngược các kí tự trong chuỗi.
Ví dụ: nhập ABCDE, xuất ra màn hình là: EDCBA
- Nhập vào chuỗi s1 và s2, cho biết vị trí xuất hiện của chuỗi s2 trong s1.
- Viết 1 chương trình đếm một ký tự xuất hiện bao nhiêu lần trong chuỗi.(vd:xâu a nhập là “ho chi minh”, nhập ‘i’ =>kq: 2)
-