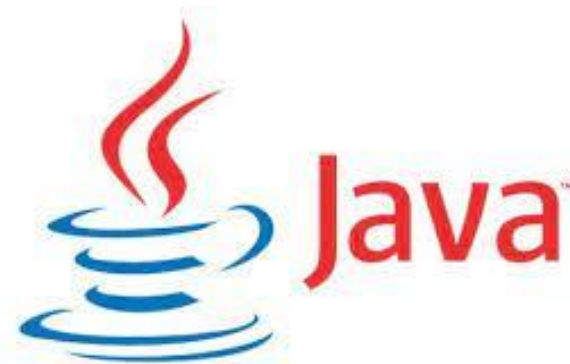


Chương 02: GIỚI THIỆU VỀ JAVA

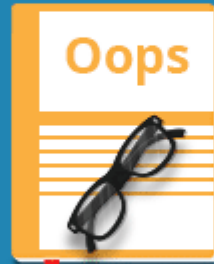


GVHD: ThS. Hồ Khôi

Email: khoihue@gmail.com



What is Java?



edureka!

Giới thiệu ngôn ngữ Java

- Java là một ngôn ngữ lập trình hướng đối tượng (**OOP** – *Object-Oriented Programming*) cho phép xây dựng các chương trình điều khiển thiết bị dựa trên các thành phần là các lớp (**class**). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành **bytecode**, bytecode sau đó sẽ được thực thi trong một môi trường độc lập (*runtime environment*), hoạt động trên hệ điều hành gọi là *máy ảo Java* (*Java Virtual Machine*).

Giới thiệu ngôn ngữ Java

- Java được phát triển bởi James Gosling và các chuyên gia của Sun Microsystems vào năm 1991. Ban đầu ngôn ngữ này được đặt tên là **Oak**, với mục tiêu thiết kế lại và loại bỏ bớt các tính năng “*nguy hiểm*” có trong ngôn ngữ C++,
- Java được tạo ra với tiêu chí "*Viết một lần, chạy ở mọi nơi*" thông qua khẩu hiệu "*Write Once, Run Anywhere*" (**WORA**). Môi trường thực thi của Sun Microsystems (*Máy ảo Java*) được thiết kế để chạy trên hầu hết các hệ điều hành nổi tiếng như: Sun Solaris, Linux, Mac OS, FreeBSD & Windows.
- Vào năm 1994, Java chính thức được phát hành cho đến nay, Java trở nên nổi tiếng là một ngôn ngữ lập trình được sử dụng trong cả lĩnh vực phát triển phần mềm và nghiên cứu, cũng như giảng dạy cho sinh viên ngành CNTT
- Giai đoạn 2009-2010, Oracle mua lại Sun Microsystems, Java đã chính thức trở thành một trong những sản phẩm chủ lực của Oracle Corporation

Mục tiêu thiết kế Java

- **Safety**, Người dung hoàn toàn yên tâm khi tải các ứng dụng Java thông qua WWW
- **Portable**, Có thể phát triển sản phẩm trên 1 hệ thống, nhưng hoàn toàn có thể chạy trên các hệ thống khác
- **Distributed**, Người dung thông thường có thể khai thác & sử dụng các dịch vụ mạng khác nhau
- **Scaleable**, để phát triển các ứng dụng thực tế bằng cách mở rộng thư viện các lớp đã xây dựng trước đó

Một số đặc điểm

- Java là ngôn ngữ hoàn toàn hướng đối tượng. Tất cả các chương trình đều gọi & sử dụng đối tượng trong quá trình hoạt động.
- Các chương trình Java luôn được biên dịch thành **bytecode**, độc lập so với hệ điều hành.
- Các **bytecode** của Java thi hành trong Máy ảo Java (**JVM**)
- Các chương trình Java có thể được xây dựng thành các “**Package**” để tạo nên các thư viện mã nguồn, cung cấp nhiều giải pháp đa dạng & linh hoạt cho các nhu cầu thực tế của ứng dụng.

JVM - Java Virtual Machine

- Máy ảo Java cung cấp môi trường thực thi các bytecode, mã thực thi được biên dịch từ các Java classes
- Các nhiệm vụ chính của JVM bao gồm:
 - ✓ Loads code
 - ✓ Verifies code
 - ✓ Executes code
 - ✓ Provides runtime environment
- Các thành phần JVM cung cấp, bao gồm:
 - ✓ Memory area
 - ✓ Class file format
 - ✓ Register set
 - ✓ Garbage-collected heap
 - ✓ Fatal error reporting etc.

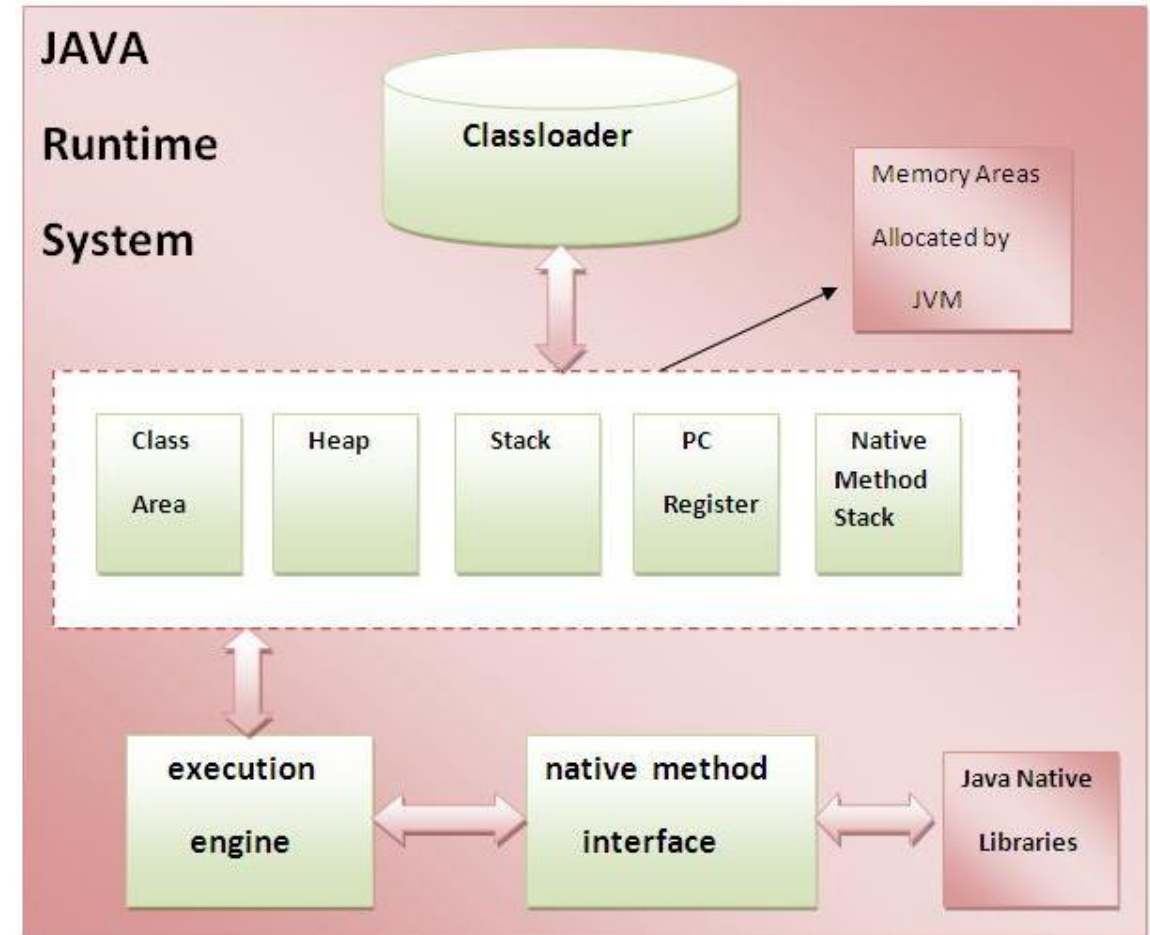


Image link: <https://www.javatpoint.com/images/jvminternal.JPG>

JRE - Java Runtime Environment

- JRE cung cấp môi trường thực thi, bao gồm cả việc triển khai JVM. Các thành phần mà JRE cung cấp là tập các thư viện mã nguồn đã được biên dịch, cùng các tài nguyên cần thiết để cho JVM sử dụng trong quá trình hoạt động

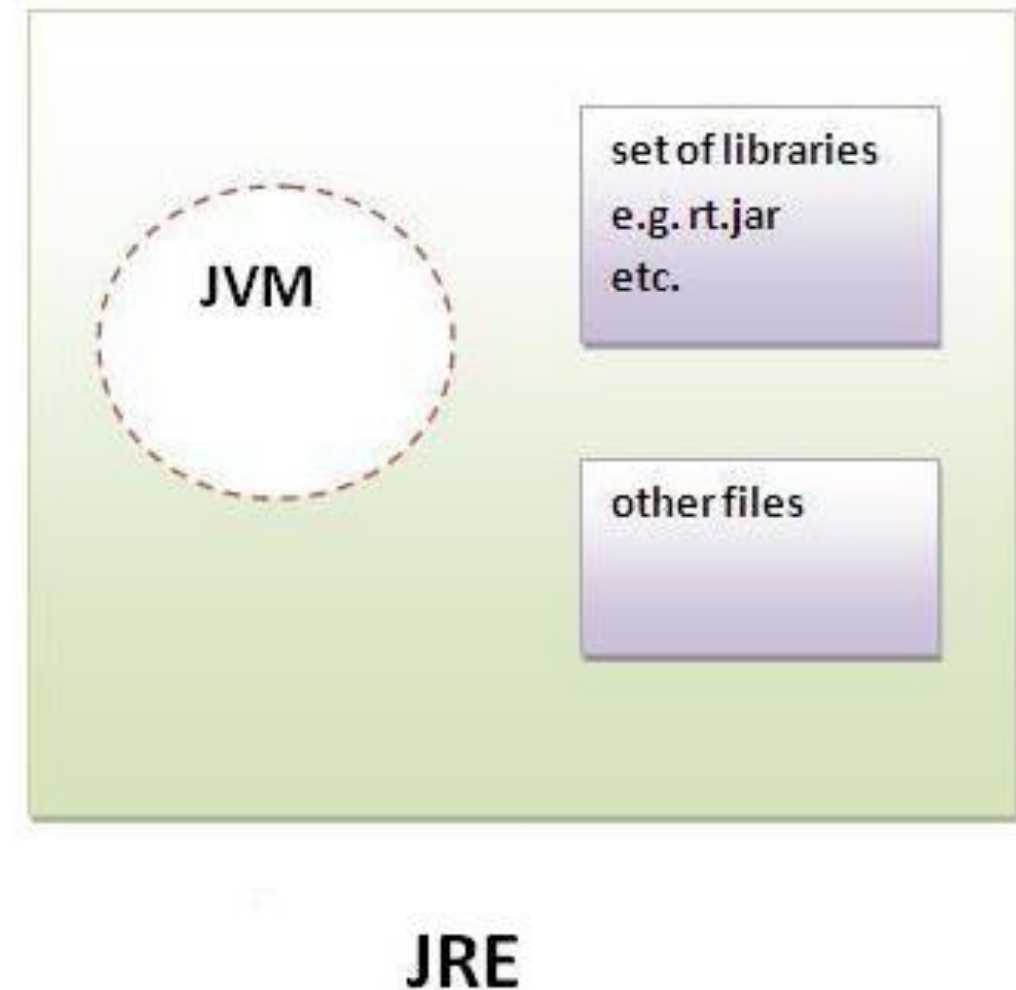


Image link: <https://www.javatpoint.com/images/jre2.JPG>

JDK – Java Development Kit

- Bộ phát triển phần mềm Java, JDK bao gồm JRE và các công cụ cần thiết khác phục vụ cho mục tiêu phát triển ứng dụng bằng ngôn ngữ Java

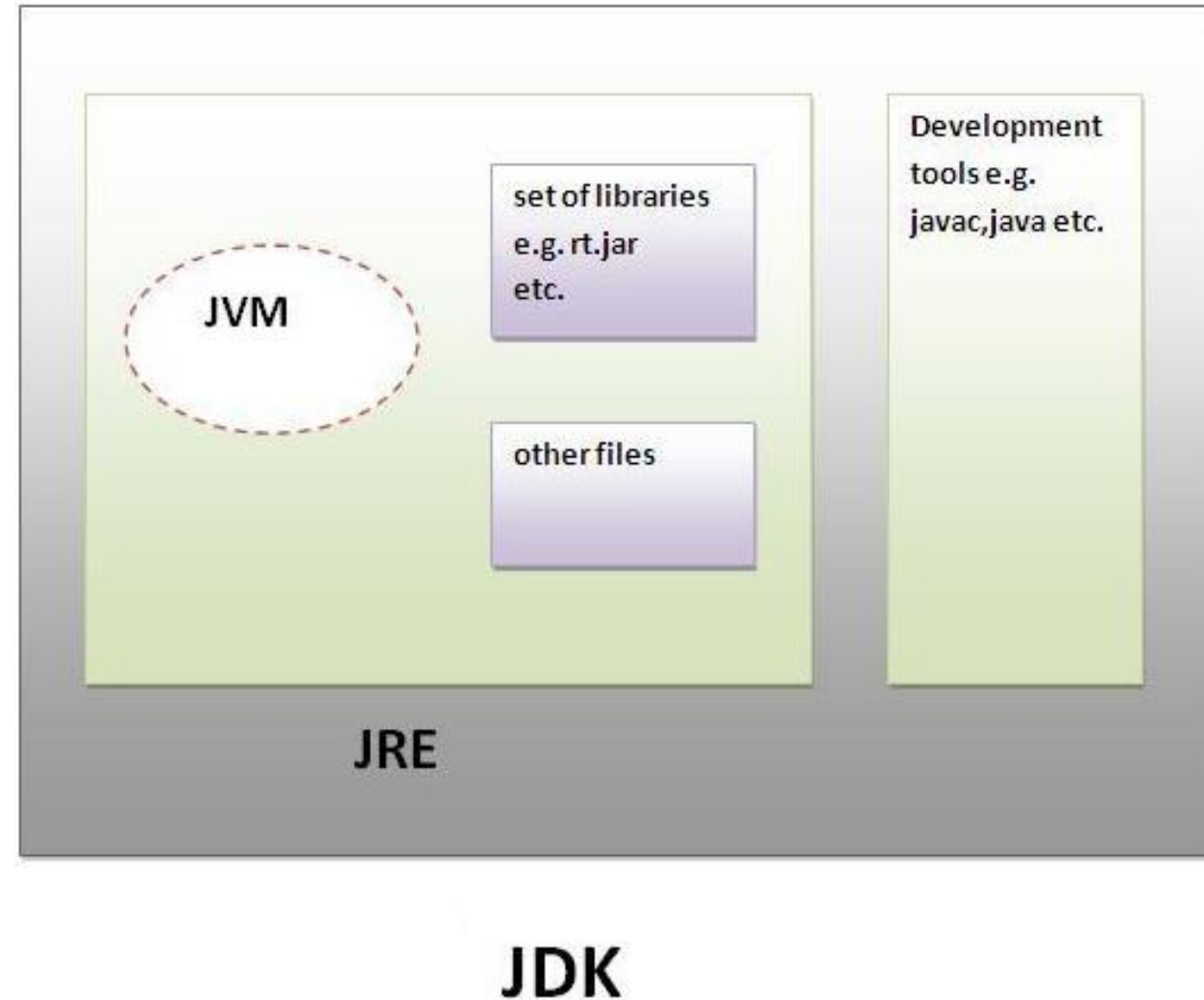


Image link: <https://www.javatpoint.com/images/jdk2.JPG>

Các dạng sản phẩm

- **Console applications**

- *Chỉ đơn thuần là xử lý & giao tiếp với người dùng dựa trên text*

- **Applets**

- *Hoạt động trên các trang web, không còn phổ biến từ những năm 2005, hạn chế về bảo mật & tốn tài nguyên khi trình duyệt nạp trang web*

- **Frame-Based Applications**

- *Desktop application*

- **Servlets**

- *Các chương trình chạy trên Web servers*

Một số quy ước

- Mỗi Java class nên lưu trữ trong một tập tin riêng biệt có phần mở rộng là “**.java**”
- Tên của tập tin “**.java**” nên trùng tên với class name đã được định nghĩa trong mã nguồn
- Tất cả các tập tin mã nguồn của một chương trình nên được quản lý bởi các Directory / Folder có cấu trúc “*phù hợp*”

Các thành phần một chương trình Java

-  Chú giải - Comments
-  Đóng gói - Package
-  Từ khóa - Reserved words
-  Từ bổ nghĩa - Modifiers
-  Câu lệnh - Statements
-  Khối - Blocks
-  Lớp - Classes
-  Phương thức - Methods
-  Phương thức chính - The main method

Comments

❑ Trong Java, các chú giải có thể được đặt :

- sau 2 dấu gạch chéo // trên 1 dòng
- giữa dấu mở /* và đóng */ trên 1 hoặc nhiều dòng




❑ Khi trình biên dịch gặp:

- //, nó bỏ qua tất cả các ký tự sau // trên dòng đó
- /*, nó quét tìm đến */ tiếp sau và bỏ qua mọi ký tự nằm giữa /* và */.





Package

- Dòng thứ hai trong chương trình (`package ch01;`) xác định một tên gói, `ch01`, cho class `Welcome`.
- Forte biên dịch source code trong tệp `Welcome.java`, tạo ra tệp `Welcome.class`, và lưu `Welcome.class` trong thư mục `ch01`.




Reserved Words

-  *Reserved words* hay *keywords* là những từ có nghĩa xác định đối với trình biên dịch và không thể sử dụng cho các mục đích khác trong chương trình.
-  VD: khi trình biên dịch gặp từ `class`, nó hiểu rằng từ ngay sau `class` là tên của class.
-  Các từ khóa khác trong ví dụ 1.1 là `public`, `static`, và `void`. Chúng sẽ được giới



Modifiers (Từ bổ nghĩa)

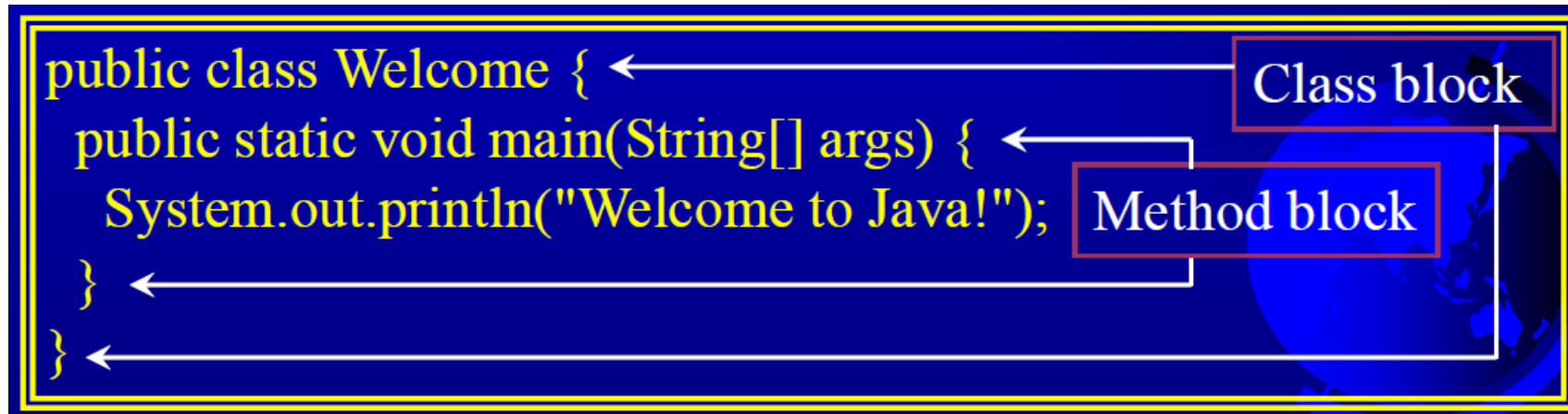
-  Java sử dụng một số từ khóa gọi là *modifiers* để xác định các thuộc tính của dữ liệu, các phương thức, lớp, và chúng có thể được sử dụng như thế nào.
-  Các ví dụ từ bổ nghĩa là public, static, private, final, abstract, và protected.
-  Một dữ liệu, phương thức, hoặc lớp public thì có thể truy nhập được bởi chương trình khác. Một dữ liệu hay phương thức private thì không thể.
-  Modifiers sẽ được thảo luận ở Chương 6, "Objects and Classes."

Statements





-  Một câu lệnh (*statement*) đại diện cho một hành động hoặc một chuỗi các hành động.
-  Câu lệnh `System.out.println("Welcome to Java!")` trong chương trình ví dụ 1.1 là một câu lệnh hiển thị lời chào "Welcome to Java!".
-  **Mọi câu lệnh trong Java kết thúc bởi một dấu chấm phẩy (;).**

Blocks





-  Một cặp dấu ngoặc nhọn trong một chương trình hình thành một khối nhóm các thành phần của một chương trình.
-  Vai trò tương tự cặp từ khóa Begin ...end; trong Pascal






Classes

-  *Class (lớp)* là thiết yếu trong xây dựng cấu trúc Java. Một class là một khuôn mẫu hay bản thiết kế cho các đối tượng.
-  Để lập trình trong Java, bạn phải hiểu các class và có thể viết, sử dụng chúng.
-  Những bí ẩn của class sẽ tiếp tục được khám phá dần xuyên suốt khóa học.
-  Bây giờ bạn chỉ cần hiểu một chương trình được xác định bằng cách sử dụng một hay nhiều class.

Methods

-  `System.out.println` là gì? Đó là một *method* (*phương thức*): một tập các câu lệnh thực hiện một chuỗi các thao tác để hiển thị một thông tin trên màn hình.
-  Nó thậm chí có thể được sử dụng mà không cần hiểu đầy đủ chi tiết nó làm việc như thế nào.
-  Nó được sử dụng bằng cách gọi một câu lệnh với tham số chuỗi ký tự (string) được bao bởi cặp dấu nháy kép. Trong trường hợp này, tham số là "Welcome to Java!"
-  Bạn có thể gọi phương thức `println` với các tham số khác nhau để in ra những message khác nhau.

main Method

-  main method cung cấp sự kiểm soát luồng chương trình. Trình biên dịch Java thực hiện ứng dụng bằng cách gọi đến main method.
-  Mọi chương trình Java phải có main method, nó là điểm khởi đầu khi thực hiện chương trình.
-  Dạng thức của main method:

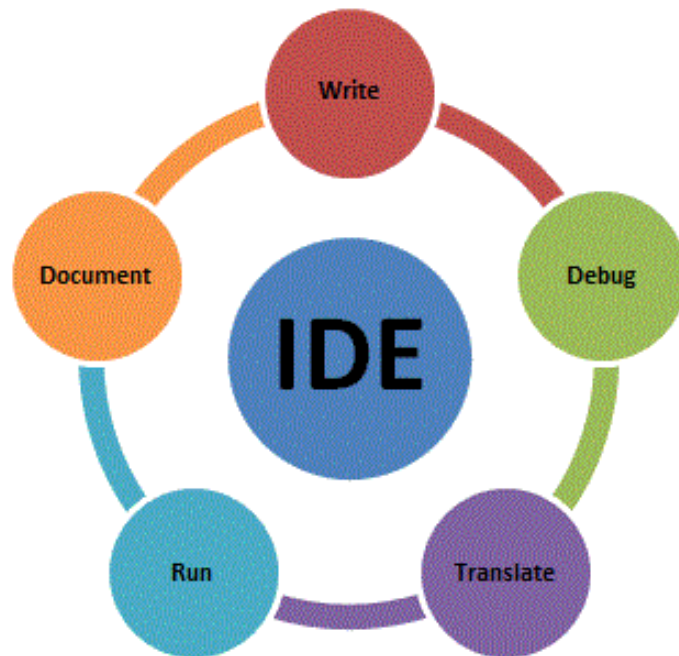
```
public static void main(String[] args) {  
  
    // Statements;  
  
}
```

Hiển thị văn bản trong Message Dialog Box

- bạn có thể sử dụng phương thức **showMessageDialog** trong lớp **JOptionPane**.
- **JOptionPane** là một trong nhiều lớp được định nghĩa trước trong hệ thống Java để có thể tái sử dụng.

IDE

(Integrated Development Environment)



Setup JDK & Eclipse

- **JDK**

(<https://www.oracle.com/java/technologies/downloads/>)

- **Eclipse IDE**

(<http://www.eclipse.org/downloads/>)



- **NetBeans IDE**

(<https://netbeans.org/downloads/>)



- **IntelliJ IDEA**

(<https://www.jetbrains.com/idea/download/#section=windows>)



Một số thuật ngữ

- **J2SE** (*Java 2 Standard Edition*) là một nền tảng thực thi (*bao gồm cả phát triển ứng dụng và triển khai*) cho các ứng dụng Java. **J2SE** cung cấp các API, các kiến trúc chuẩn, các thư viện lớp và các công cụ cần thiết để xây các ứng dụng Java. **J2SE** vẫn được xem là nền tảng thiên về phát triển các sản phẩm chạy trên máy tính để bàn. **J2SE** gồm 2 thành phần chính: JRE & JDK

(https://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html#A1096936)

- **JRE** – Java 2 Runtime Environment
- **JDK** – Java 2 Software Development Kit

Một số thuật ngữ

- **J2EE** (Java 2 Platform, Enterprise Edition, hay Java EE) là một nền tảng cho phép lập trình để phát triển ứng dụng phân tán trên kiến trúc đa tầng (nLayer), chủ yếu dựa vào các module chạy trên các máy chủ ứng dụng.
- **J2ME** được phát triển từ kiến trúc Java Card, Embedded Java và Personal Java của phiên bản Java 1.1. Đến sự ra đời của Java 2 thì Sun quyết định thay thế Personal Java và được gọi với tên mới là Java 2 Micro Edition, hay viết tắt là **J2ME**. Đúng với tên gọi, J2ME là nền tảng cho các thiết bị có tính chất nhỏ gọn. J2ME is a computing platform for development and deployment of portable code for embedded and mobile devices (*micro-controllers, sensors, gateways, mobile phones, personal digital assistants, TV set-top boxes, printers*)

Cấu trúc chương trình Java

```
package com.poly;  
public class Program{  
    public static void main(String[] args){  
        // mã thực thi  
    }  
}
```

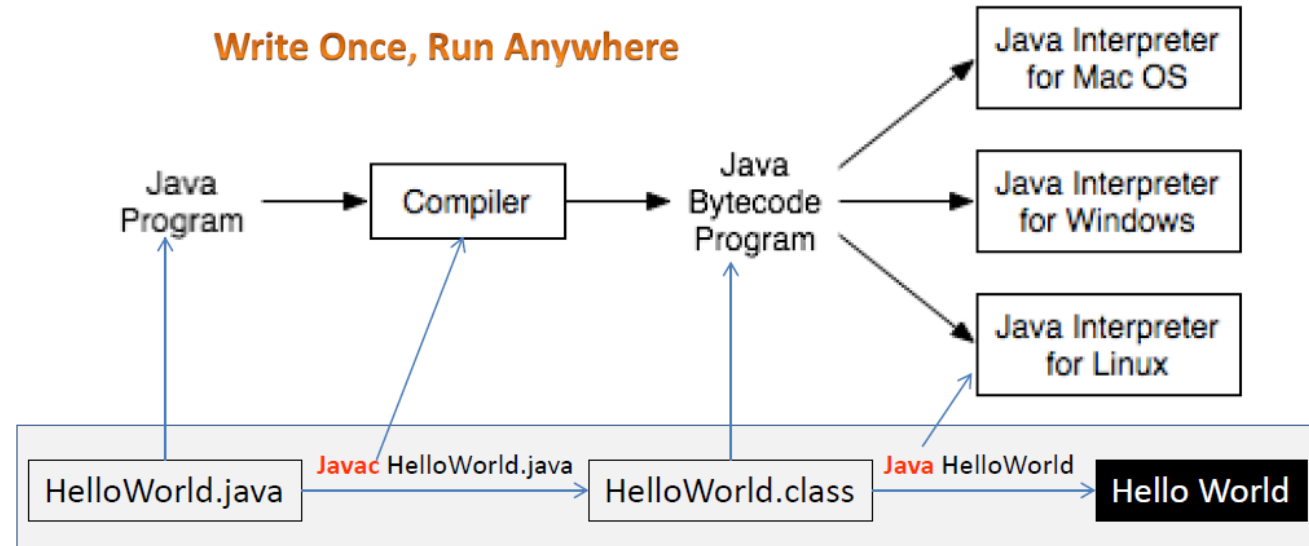
Save as

Program.java

- ❑ **com.poly**: tên gói chứa lớp
 - ❖ Sử dụng ký tự thường và dấu chấm. Có thể xem package như folder còn class như file.
- ❑ **Program**: tên lớp
 - ❖ Phải giống tên file java. Viết hoa ký tự đầu của mỗi từ
- ❑ **main()**: phương thức bắt đầu chạy
 - ❖ Lớp có thể có nhiều phương thức nhưng main() được gọi tự động khi ứng dụng chạy

BYTECODE

- ❑ Khác với ngôn ngữ lập trình khác, thay vì biên dịch mã nguồn thành mã máy, Java được thiết kế biên dịch mã nguồn thành bytecode
- ❑ Bytecode sau đó được môi trường thực thi chạy



JDK – JAVA DEVELOPMENT KIT

- ❑ JDK và các công cụ (javac, java)
- ❑ Cấu hình JDK (path, classpath)

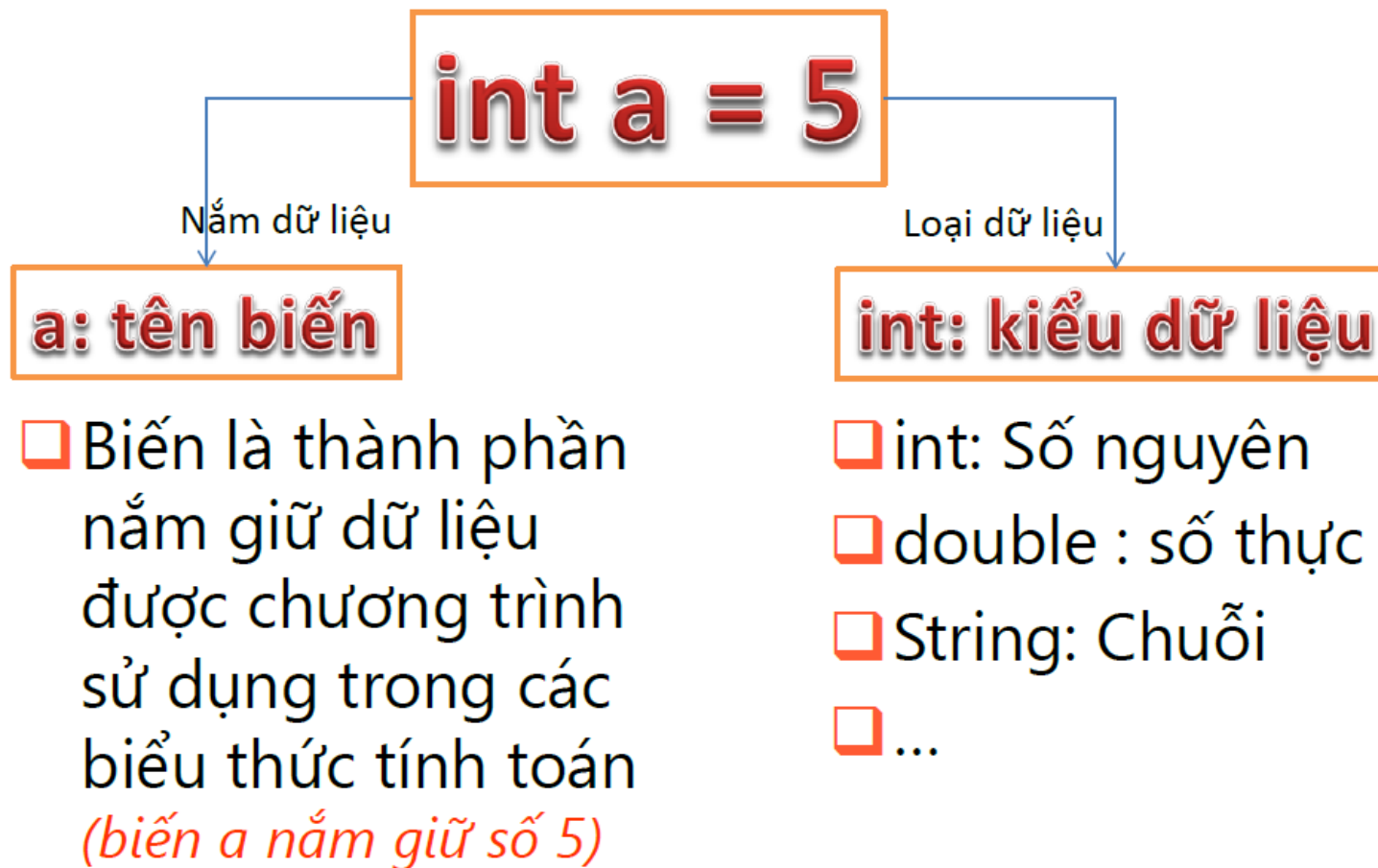


Giới thiệu Java IDE

Hỗ trợ việc phát triển và triển khai ứng dụng dễ dàng hơn



KHÁI NIỆM BIẾN



KHAI BÁO BIẾN

❑ Cú pháp

<kiểu dữ liệu> <tên biến> [=giá trị khởi đầu];

❑ Ví dụ:

int a; // khai báo biến không khởi đầu giá trị

double b = 5; // khai báo biến có khởi đầu giá trị

❑ Khai báo nhiều biến cùng kiểu

int a, b=5, c;

❑ Gán giá trị cho biến

c = 9;

a = 15;

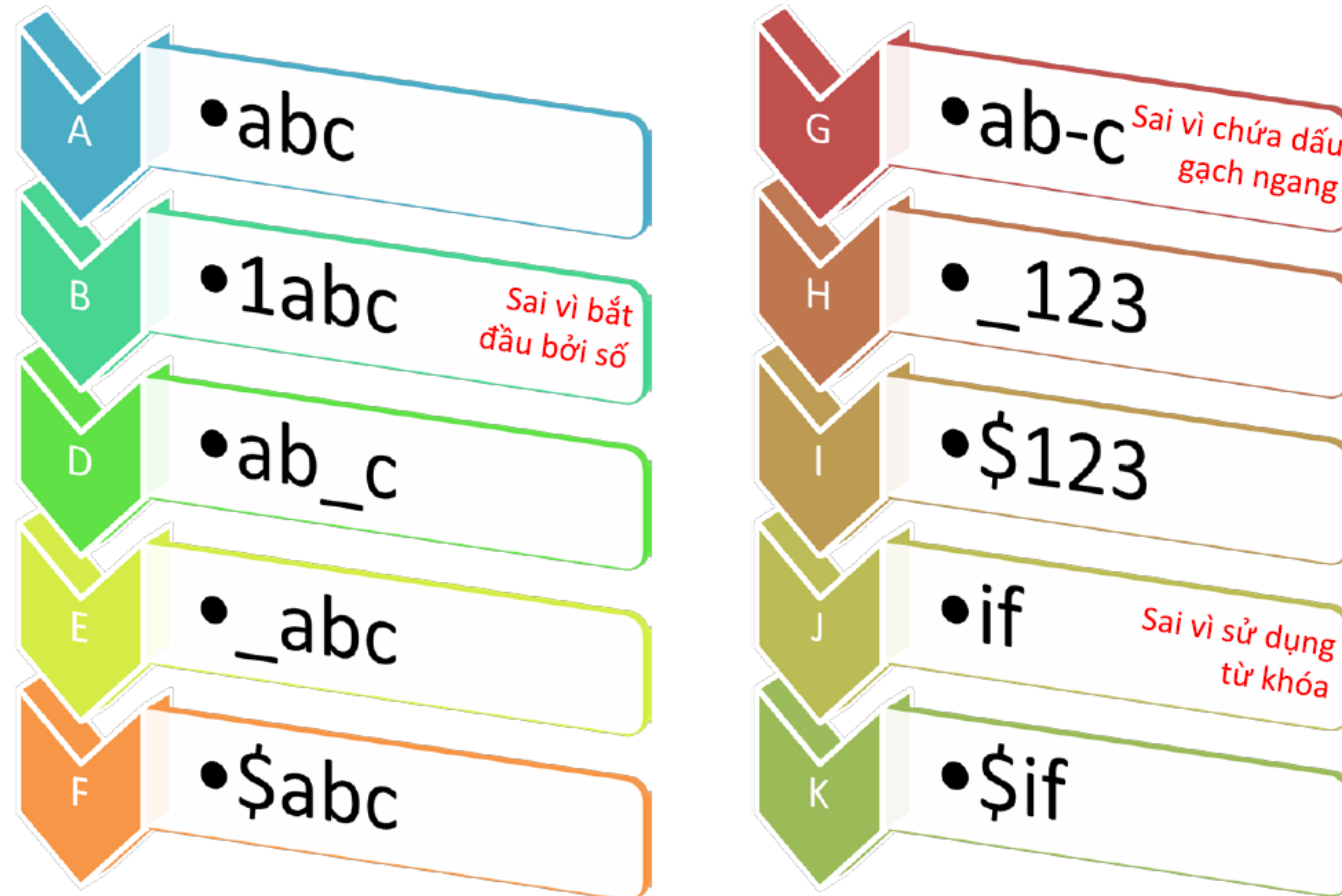
ĐẶT TÊN BIẾN

- ❑ Sử dụng ký tự **alphabet**, **số**, **\$** hoặc gạch dưới (**_**).
Tên có phân biệt **HOA/thường**
- ❑ Không ~~bắt đầu bởi số~~, không dùng ~~từ khóa~~

** Từ khóa là các từ được sử dụng để xây dựng ra ngôn ngữ lập trình java*

abstract	assert	boolean	break	byte	case
catch	char	class	const	continue	default
do	double	else	enum	extends	final
finally	float	for	goto	if	implements
import	instanceof	int	interface	long	native
new	package	private	protected	public	return
short	static	strictfp	super	switch	synchronized
this	throw	throws	transient	try	void
volatile	while				

TÊN BIẾN NÀO SAU ĐÂY KHÔNG HỢP LỆ



CÁC HÀM XUẤT RA MÀN HÌNH

- ❑ `System.out.print()`: Xuất xong không xuống dòng
- ❑ `System.out.println()`: Xuất xong có xuống dòng
- ❑ `System.out.printf()`: Xuất có định dạng, các ký tự định dạng

NHẬP TỪ BÀN PHÍM

- ❑ java.util.Scanner cho phép nhận dữ liệu từ bàn phím một cách đơn giản
- ❑ Tạo đối tượng Scanner
 - ❖ Scanner scanner = new Scanner(System.in)
- ❑ Các phương thức thường dùng
 - ❖ scanner.**nextLine()**
 - Nhận 1 dòng nhập từ bàn phím
 - ❖ scanner.**nextInt()**
 - Nhận 1 số nguyên nhập từ bàn phím
 - ❖ scanner.**nextDouble()**
 - Nhận 1 số thực nhập từ bàn phím

CÁC HÀM TOÁN HỌC

- ❑ Java cung cấp các hàm tiện ích giúp chúng ta thực hiện các phép tính khó một cách dễ dàng như:
 - ❖ Làm tròn số
 - ❖ Tính căn bậc 2
 - ❖ Tính lũy thừa
 - ❖ ...
- ❑ Ví dụ sau đây tính căn bậc 2 của 7
 - ❖ `double a = Math.sqrt(7)`
- ❑ Ngoài `Math.sqrt()` còn rất nhiều hàm khác được trình bày ở slide sau.

CÁC HÀM TOÁN HỌC

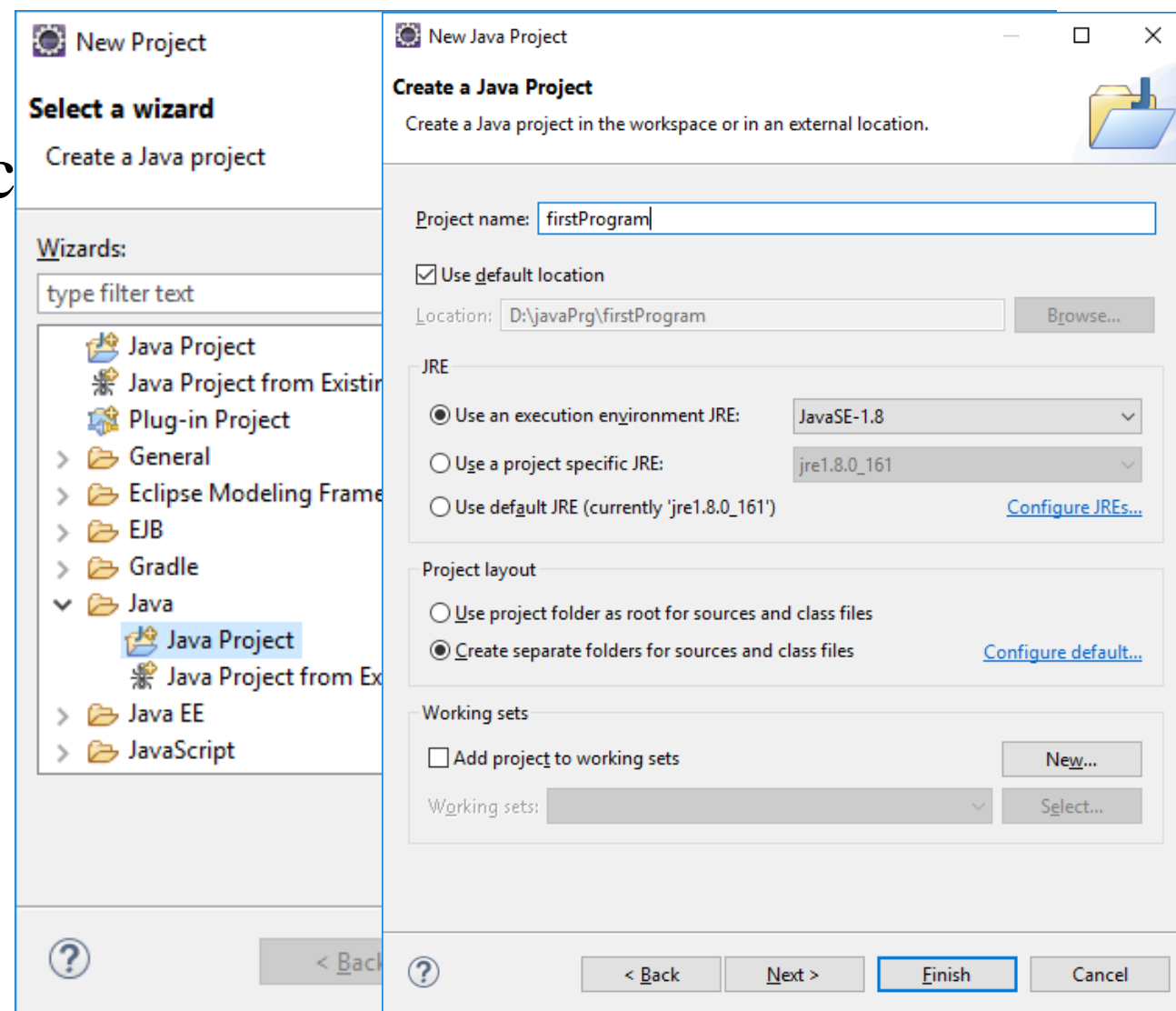
Hàm	Diễn giải	Ví dụ
Math.min(a, b)	Lấy số nhỏ nhất của 2 số a và b	$x = \text{Math.min}(5, 3.5) \Rightarrow x = 3.5$
Math.max(a, b)	Lấy số lớn nhất của 2 số a và b	$x = \text{Math.max}(5, 3.5) \Rightarrow x = 5$
Math.pow(a, n)	Tính a^n (a lũy thừa n)	$x = \text{Math.pow}(5, 3) \Rightarrow x = 75$
Math.sqrt(a)	Tính \sqrt{a} (căn bậc 2 của a)	$x = \text{Math.sqrt}(16) \Rightarrow x = 4$
Math.abs(a)	Lấy giá trị tuyệt đối của a	$x = \text{Math.abs}(-5) \Rightarrow x = 5$
Math.ceil(a)	Lấy số nguyên trên của a	$x = \text{Math.ceil}(3.5) \Rightarrow x = 4$
Math.floor(a)	Lấy số nguyên dưới của a	$x = \text{Math.floor}(3.5) \Rightarrow x = 3$
Math.round(a)	Làm tròn số của a	$x = \text{Math.round}(3.5) \Rightarrow x = 4$
Math.random()	Sinh số ngẫu nhiên từ 0 đến 1	$x = \text{Math.random()} \Rightarrow x = 0..1$

First program



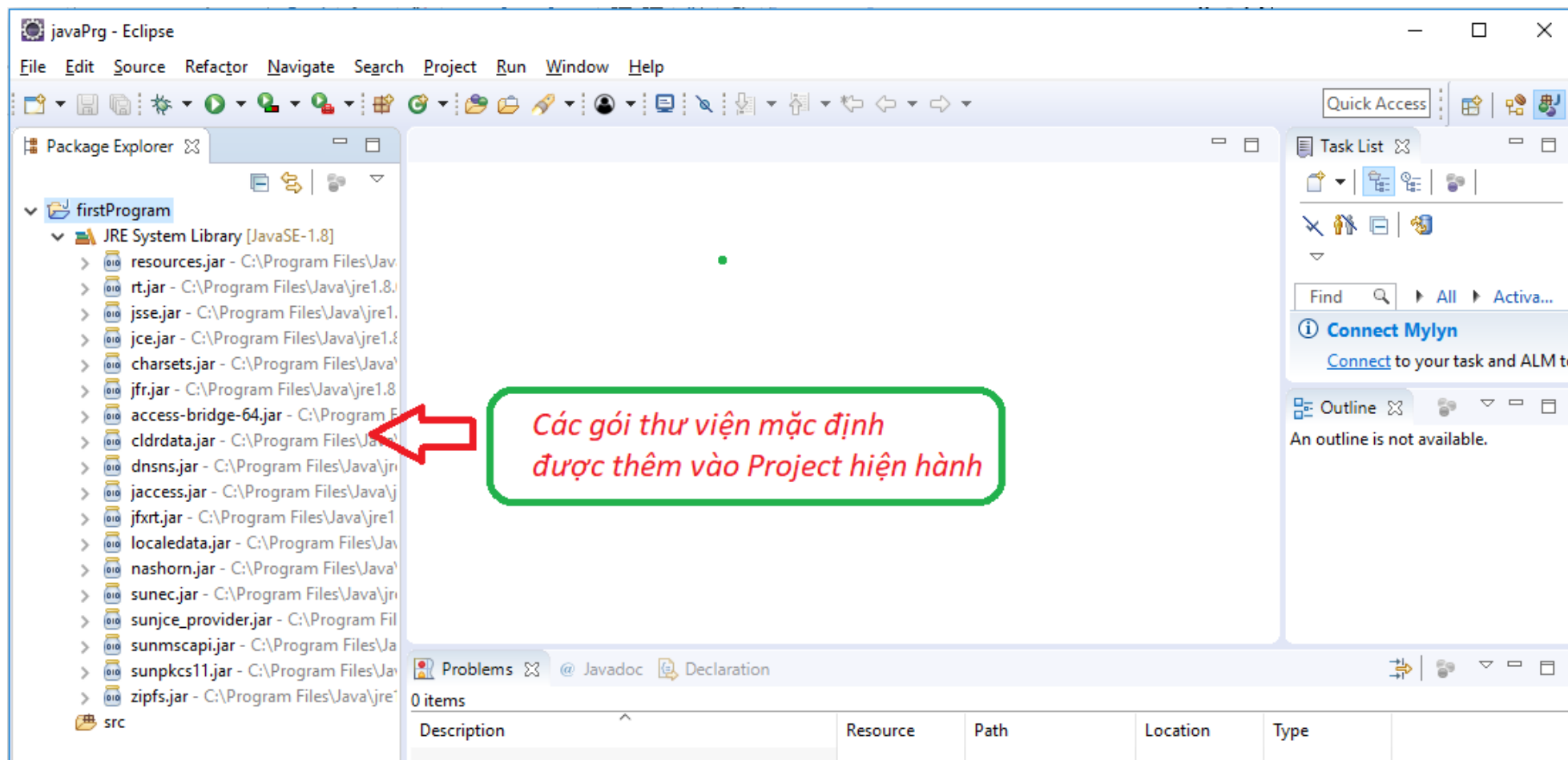
Ví dụ

- **Tạo Java project**
[Menu] File \ New -> Project



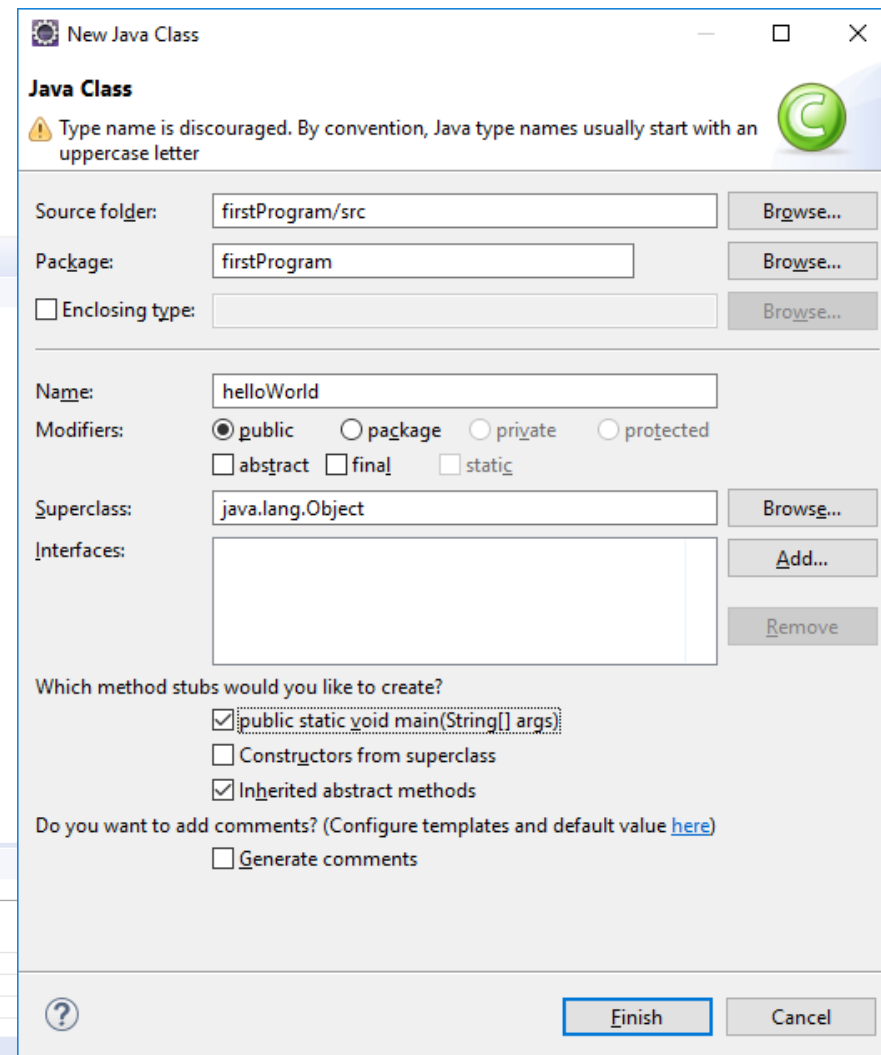
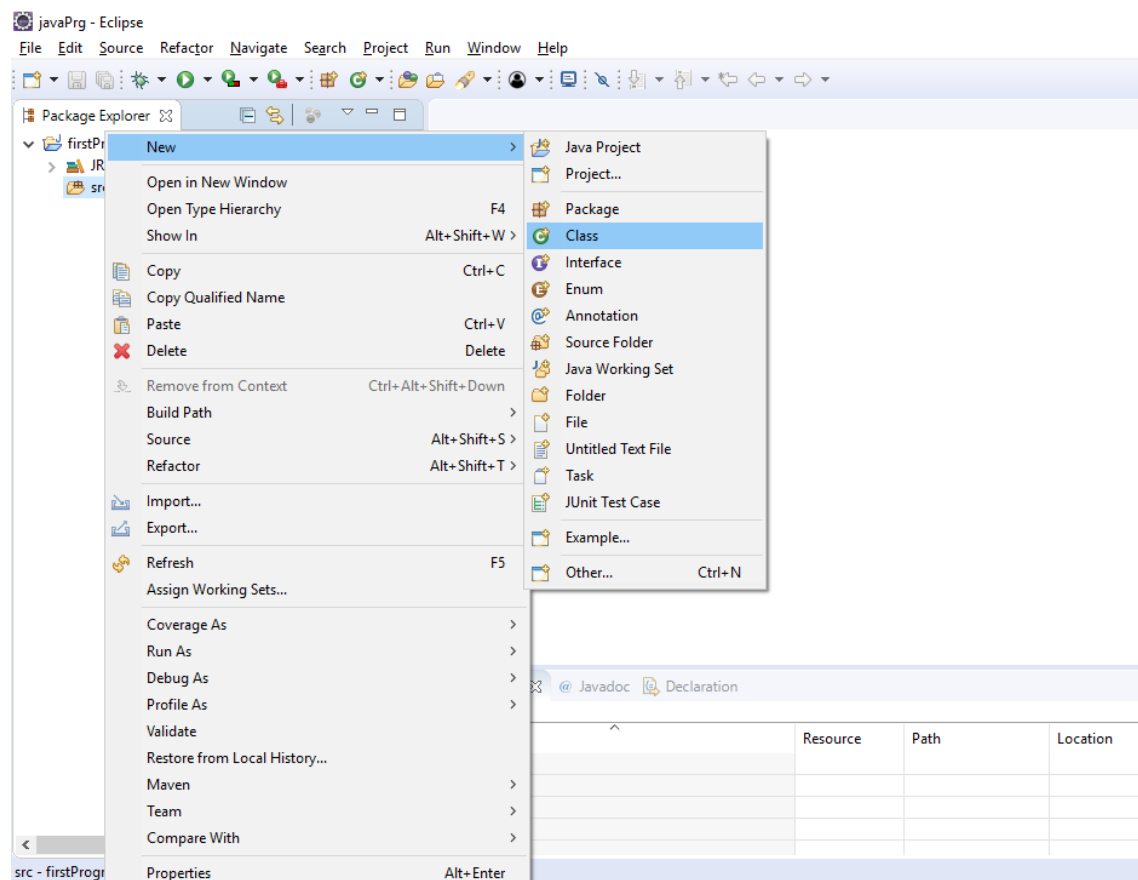
Ví dụ

- Package explorer & các thư viện mặc định



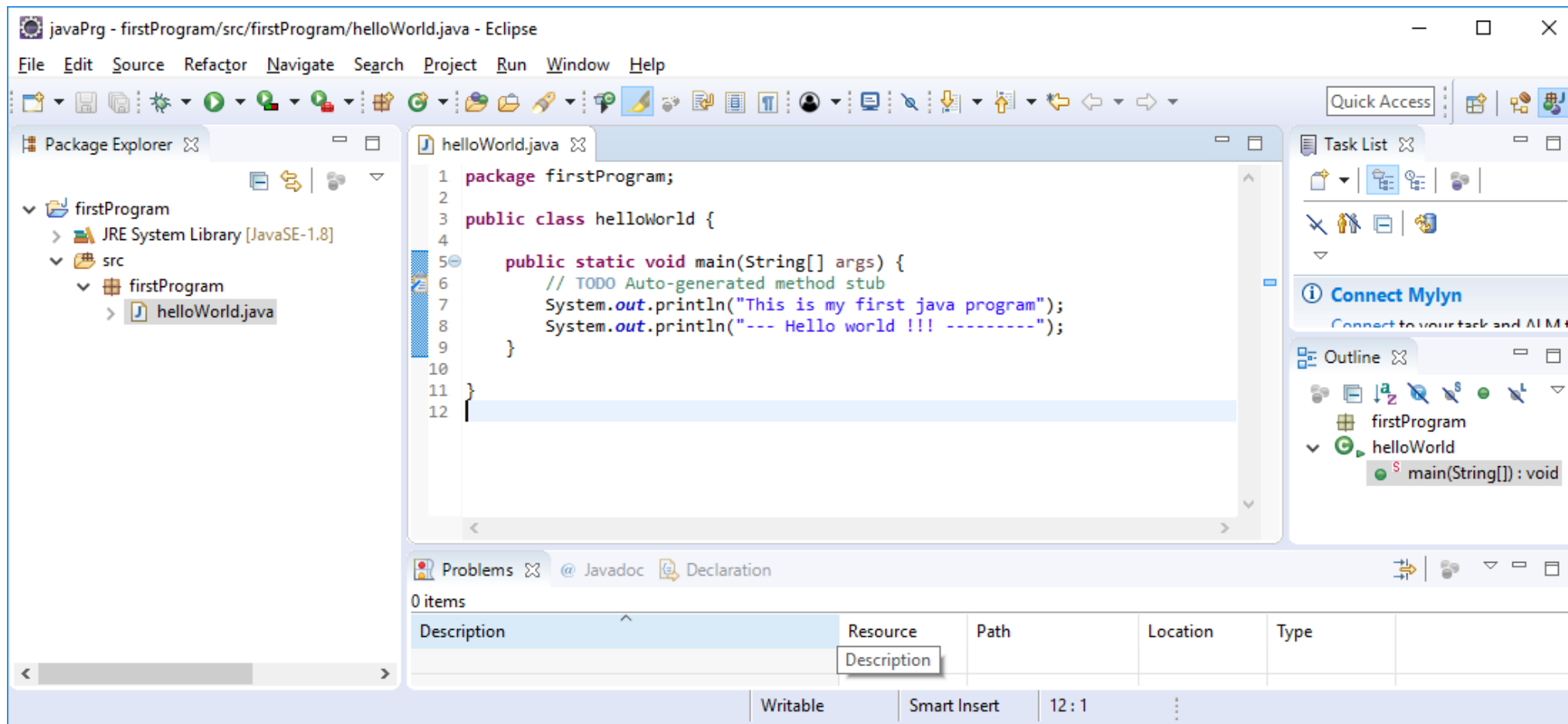
Ví dụ

- **Tạo class đầu tiên cho chương trình**
right mouse -> thư mục src, chọn new -> class



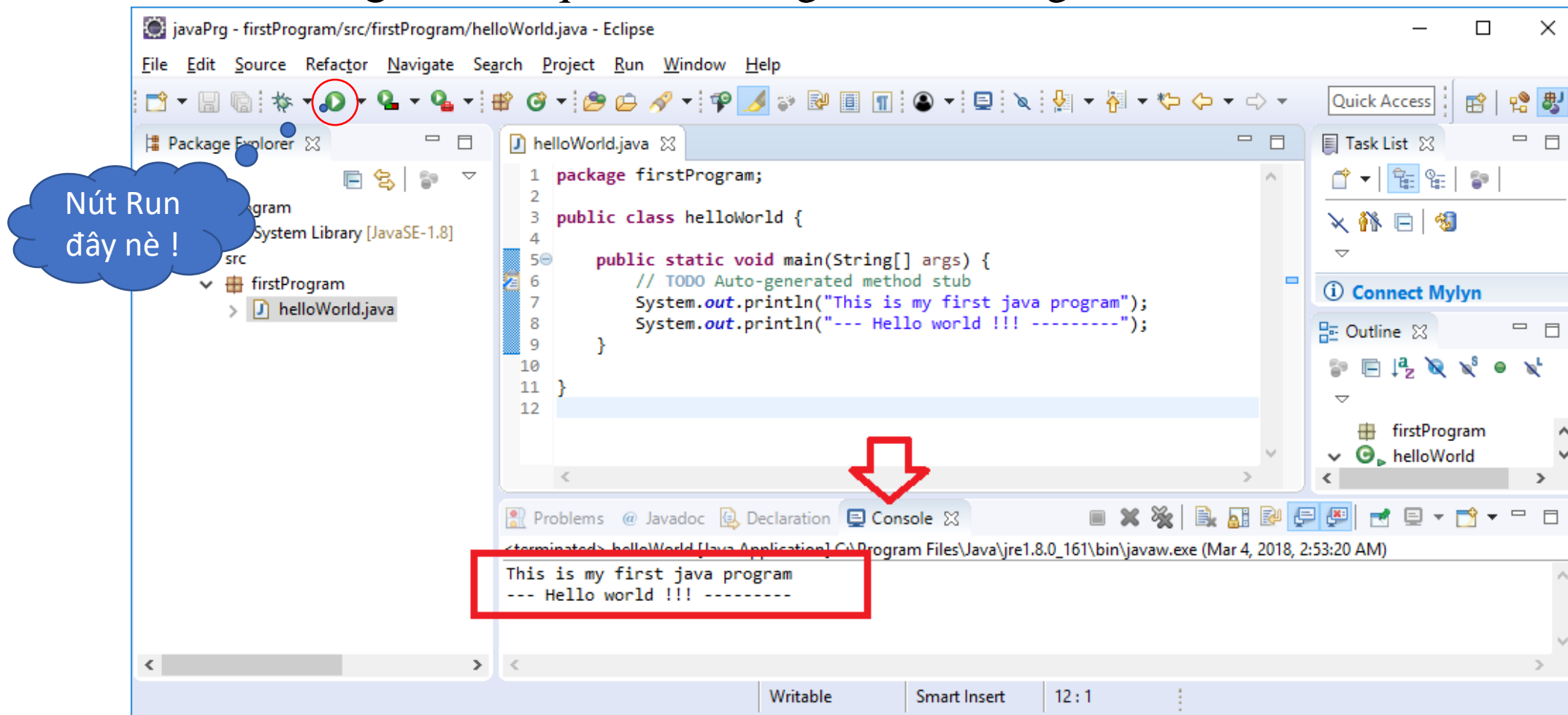
Ví dụ

- Xuất dữ liệu với **System.out.println**(“Chuỗi dữ liệu”)



Ví dụ

- Thi hành chương trình & quan sát thông tin trên vùng cửa sổ **Console**



VÍ DỤ

```
public class Sinhvien {  
    // biến static 'ten'  
    public static String ten = "Nguyen Van A";  
  
    // biến static 'tuoi'  
    public static int tuoi = 21;  
  
    public static void main(String args[]) {  
        // Sử dụng biến static bằng cách gọi trực tiếp  
        System.out.println("Ten : " + ten);  
  
        // Sử dụng biến static bằng cách gọi thông qua tên class  
        System.out.println("Ten : " + Sinhvien.tuoi);  
    }  
}
```

Bài tập

- Cho các yêu cầu để sinh viên có thể thuộc cấu trúc của một chương trình, biết cách xuất thông tin, biết cách nhập số nguyên.
- VD:
 - ✓ Viết chương trình in ra thông tin của sinh viên trên màn hình (họ tên, MSSV, ngày sinh, số điện thoại)
 - ✓ Viết chương trình in ra thông báo họp lớp
 - ✓ Viết chương trình in ra thông tin của 1 loại mặt hàng nào đó
 - ✓ Chương trình cho phép cộng, trừ, nhân, chia hai số nguyên.
 - ✓ ...

Bài tập Ví dụ 1

- ✓ Viết chương trình in ra thông tin của sinh viên trên màn hình (họ tên, MSSV, ngày sinh, số điện thoại)

```
import java.util.Scanner;

public class StudentInformation {
    public static void main(String[] args) {
        String name;
        int year;
        String address;

        Scanner scanner = new Scanner(System.in);
        System.out.println("Nhập vào họ tên của sinh viên: ");
        name = scanner.nextLine();
        System.out.println("Nhập vào năm sinh của sinh viên: ");
        year = scanner.nextInt();
        System.out.println("Nhập vào địa chỉ của sinh viên: ");
        address = scanner.nextLine();

        System.out.println("Họ tên của sinh viên: " + name);
        System.out.println("Tuổi của sinh viên = " + year);
        System.out.println("Địa chỉ của sinh viên: " + address);
    }
}
```

Bài tập Ví dụ 4

- Chương trình cho phép cộng, trừ, nhân, chia hai số nguyên.

```
package NTT;
import java.util.Scanner;

public class nhapvao2so {

    public static void main(String[] args)
    {

        int a, b, T, H, N, C;
        Scanner sc = new Scanner(System.in);
        System.out.println("Nhập vào số thứ nhất: ");
        a = sc.nextInt();

        System.out.println("Nhập vào số thứ hai: ");
        b = sc.nextInt();

        sc.close();
        T = a + b; // Tổng 2 số
        System.out.println("Tổng của hai số vừa nhập là: "+T);
        H = a - b; // Hiệu 2 số
        System.out.println("Hiệu của hai số vừa nhập là: "+H);
        N = a*b; // Tích của 2 số
        System.out.println("Tích của hai số vừa nhập là: "+N);
        C = a/b; // Thương 2 số
        System.out.println("Thương của hai số vừa nhập là: "+C);

    }

}
```


Suy ngẫm???

- Những khái niệm, thuật ngữ chính liên quan đến ngôn ngữ Java: **JVM, JRE, JDK, WORA, OOP, bytecode**
- IDE là gì ?. Tại sao lập trình viên cần đến **IDE** ?.
- Khởi động Eclipse IDE & tạo Java Project, Tạo class
- **System.out.print** hoặc **System.out.println** là các phương thức cho phép xuất dữ liệu ra màn hình của Java
- Hàm main của 1 class sẽ tự động được gọi thi hành và hàm này bắt buộc phải có khai báo “**public static void**”

Tài liệu tham khảo

- Jose M. Garrido, “**Object-Oriented Programming: From Problem Solving to Java**”
- Paul Deitel, Harvey Deitel, “**Java : How to program**”, 9th edition, 2012
- Oracle, “**The Java™ Tutorials**”,
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>, 8:10PM,
27/09/2022
- Java tutorial, <https://www.javatpoint.com/java-tutorial> , 8:10PM, 27/09/2022

Bài tập (1)

1. Viết chương trình tính chu vi và diện tích hình tròn, biết bán kính $R=15$ là một hằng số.

Hướng dẫn:

$$\text{Chu vi} = 2.PI.R$$

$$\text{Diện tích} = R^2.PI$$

2. Viết chương trình nhập hai số nguyên a, b từ bàn phím. Sau đó, in ra màn hình như sau:

Tổng hai số $a+b = ?$

Hiệu hai số $a-b = ?$

Tích hai số $a*b = ?$

3. Nhập từ bàn phím hai số thực x và y . Sau đó tính x^y và ra màn hình: x lũy thừa y bằng?

Bài tập (2)

4. Viết chương trình tính diện tích của tam giác biết ba cạnh là ba biến a,b,c được khởi tạo ban đầu là a=5.5, b=6.3, c=5.7 và ta có công thức tính diện tích.

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

và $p = (a+b+c)/2$

5. Viết chương trình tính diện tích và chu vi hình chữ nhật, với chiều dài a và chiều rộng b là số nguyên nhập từ bàn phím và in ra màn hình như sau:

Chiều dài a = ?

Chiều rộng b = ?

Diện tích hình chữ nhật s = ?

Chu vi hình chữ nhật c = ?

6. Viết chương trình tính chu vi và diện tích hình tròn, với bán kính là số thực r nhập vào từ bàn phím và in ra màn hình như sau:

Bán kính hình tròn r = ?

Chu vi hình tròn c = ?

Diện tích hình tròn s = ?

THANK YOU VERY MUCH



Q/A: khoihue@gmail.com