



# Introduce to java

Môn học: Cơ sở lập trình [*Buổi 2*]

GV: Nguyễn Mai Huy

# Java

(program language)

*Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)*

# 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*).

Nguyễn Mai Huy - [nmai.huy@boduca.com](mailto:nmai.huy@boduca.com)

# 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.

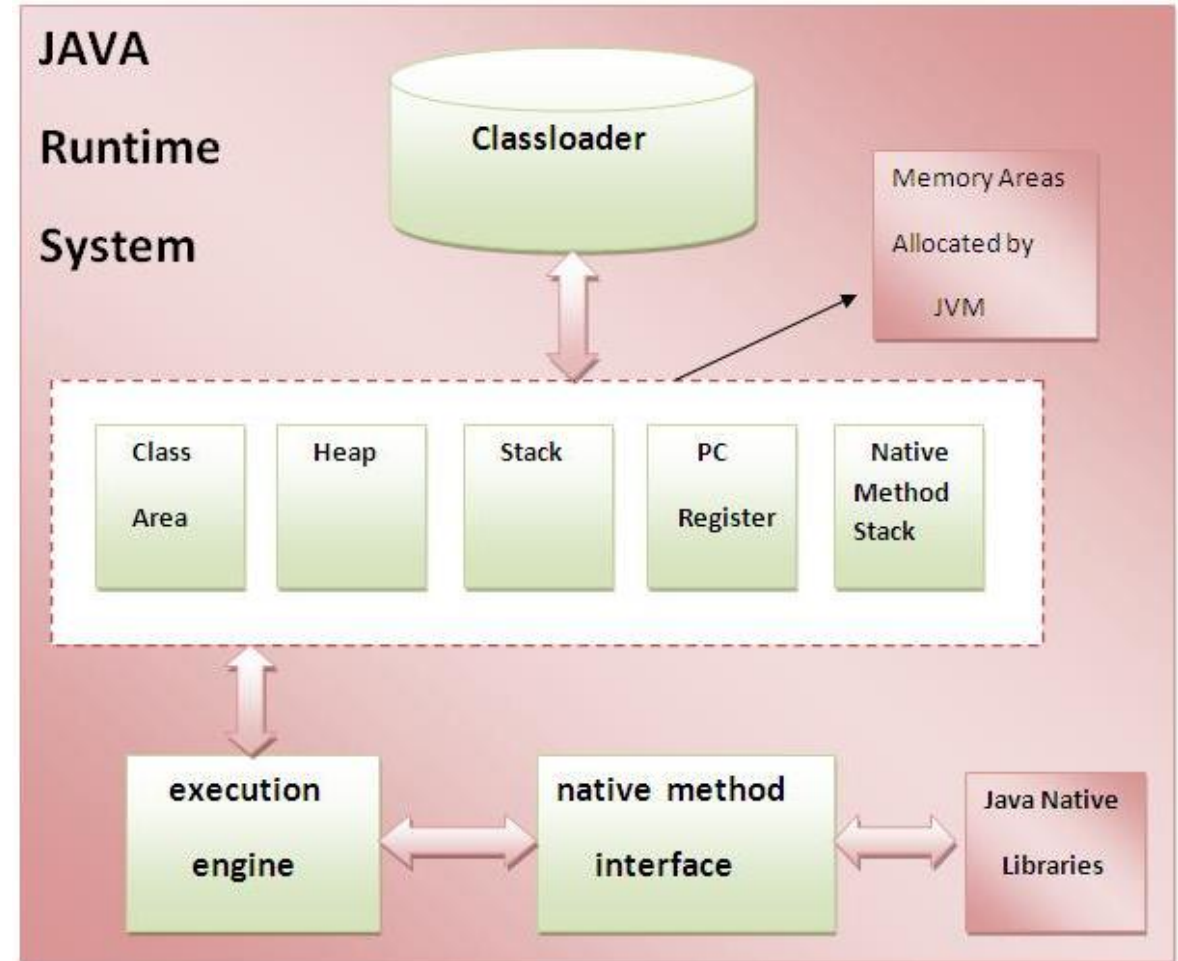


Image link: <https://www.javatpoint.com/images/jvminternal.JPG>

# 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

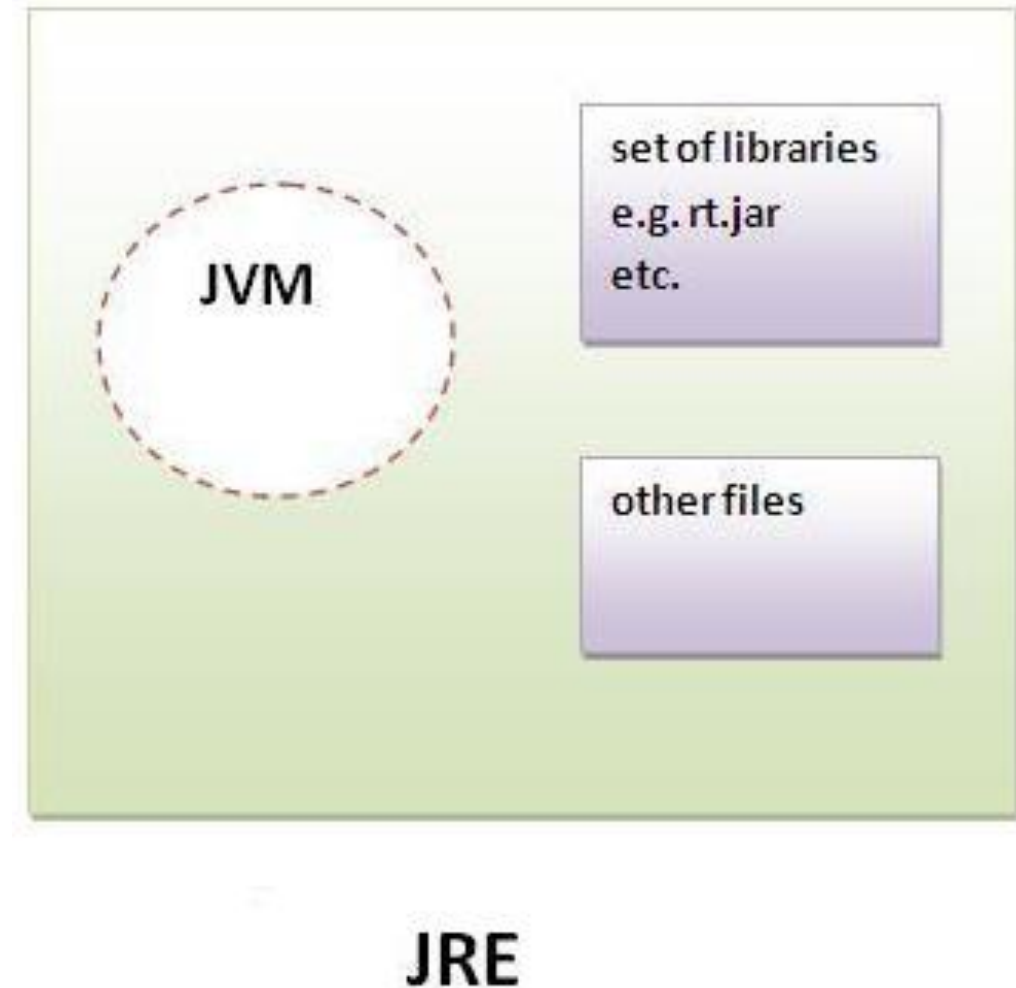


Image link: <https://www.javatpoint.com/images/jre2.JPG>



# 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

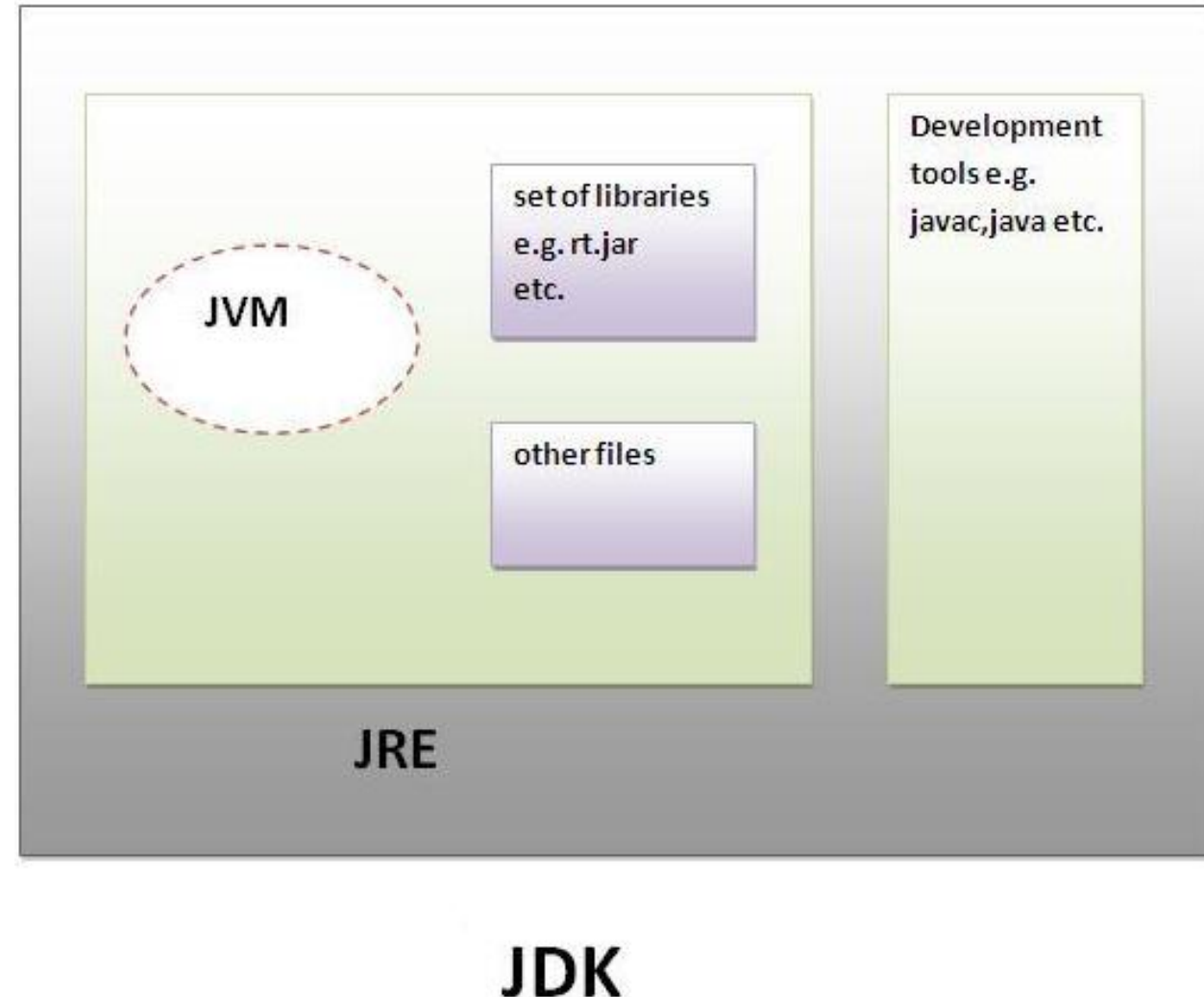


Image link: <https://www.javatpoint.com/images/jdk2.JPG>

# 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”

# IDE

(Integrated Development Environment)

*Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)*

# Setup JDK & Eclipse

- **JDK 8u161**

(<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> )

- **Eclipse IDE**

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



- **NetBeans IDE**

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





# Một số thuật ngữ

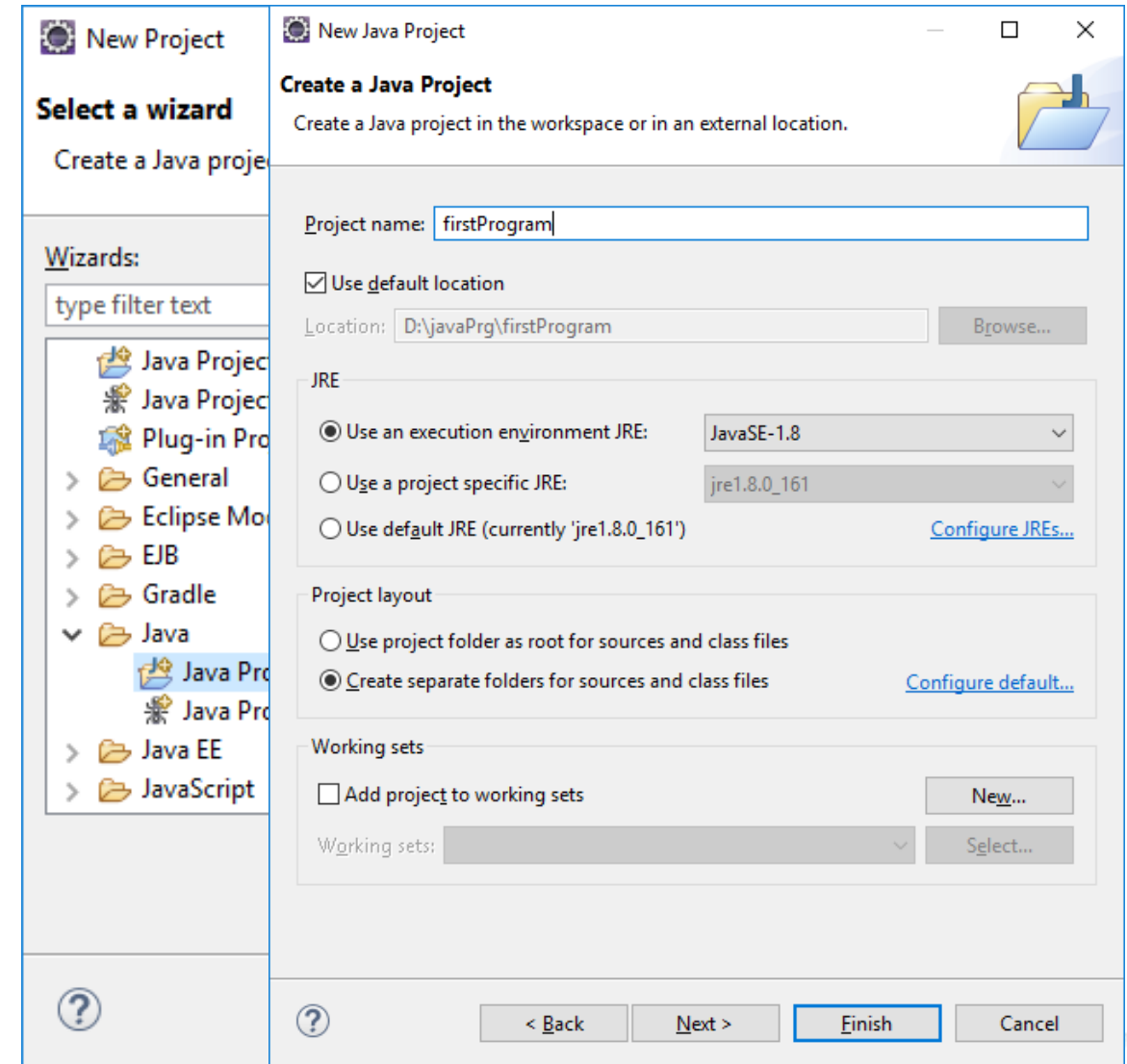
- **J2SE** (*Java 2 Standard Edition*) là một nền tảng thực thi (*bao gồm cả phát triển ứng dụng và triển khai*) cho các ứng dụng Java. **J2SE** cung cấp các API, các kiến trúc chuẩn, các thư viện lớp và các công cụ cần thiết để xây các ứng dụng Java. **J2SE** vẫn được xem là nền tảng thiên về phát triển 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](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
- **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*)

# First program

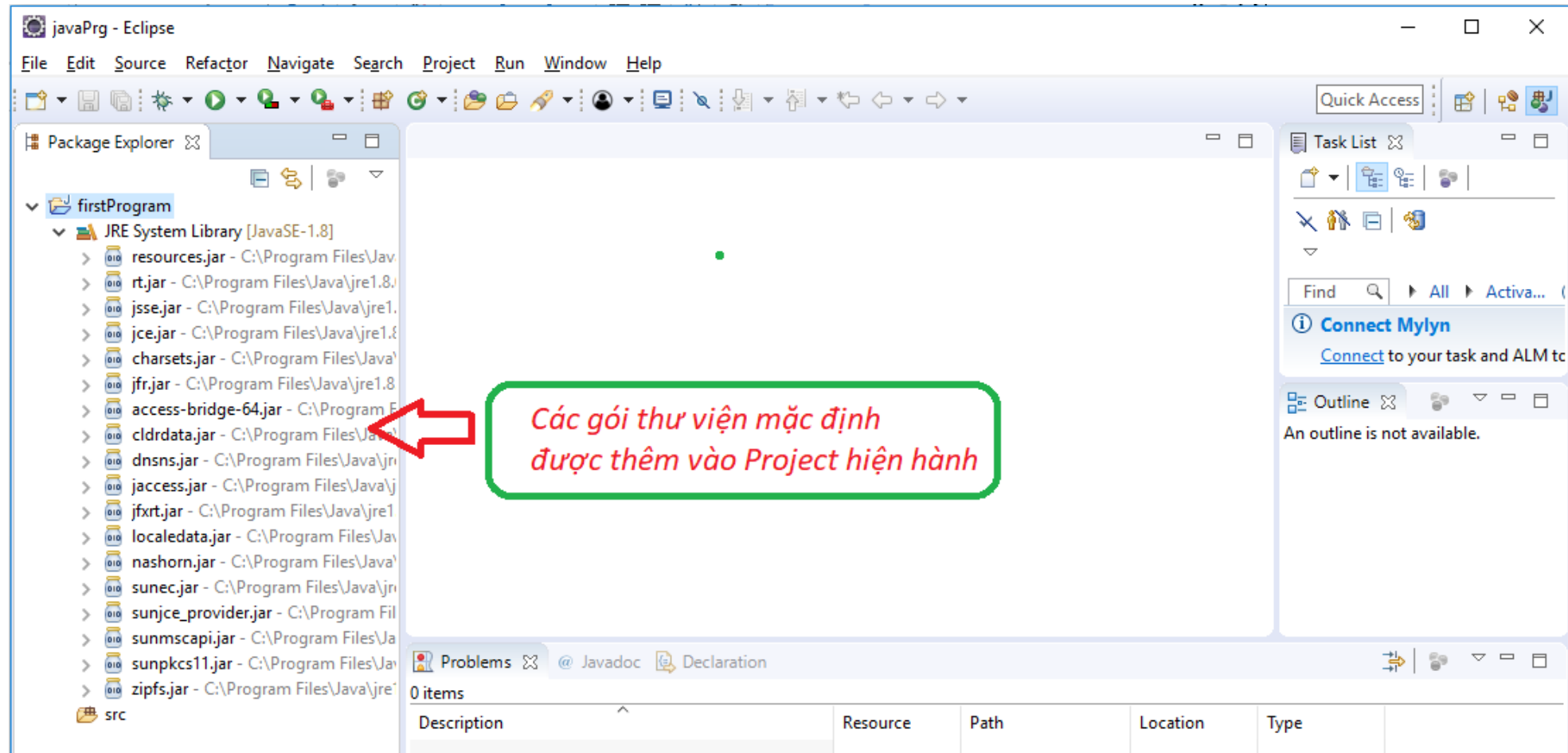
*Nguyễn Mai Huy - [nmaihuy@bodua.com](mailto:nmaihuy@bodua.com)*

# ■ Tạo Java project

[Menu] File \ New -> Project

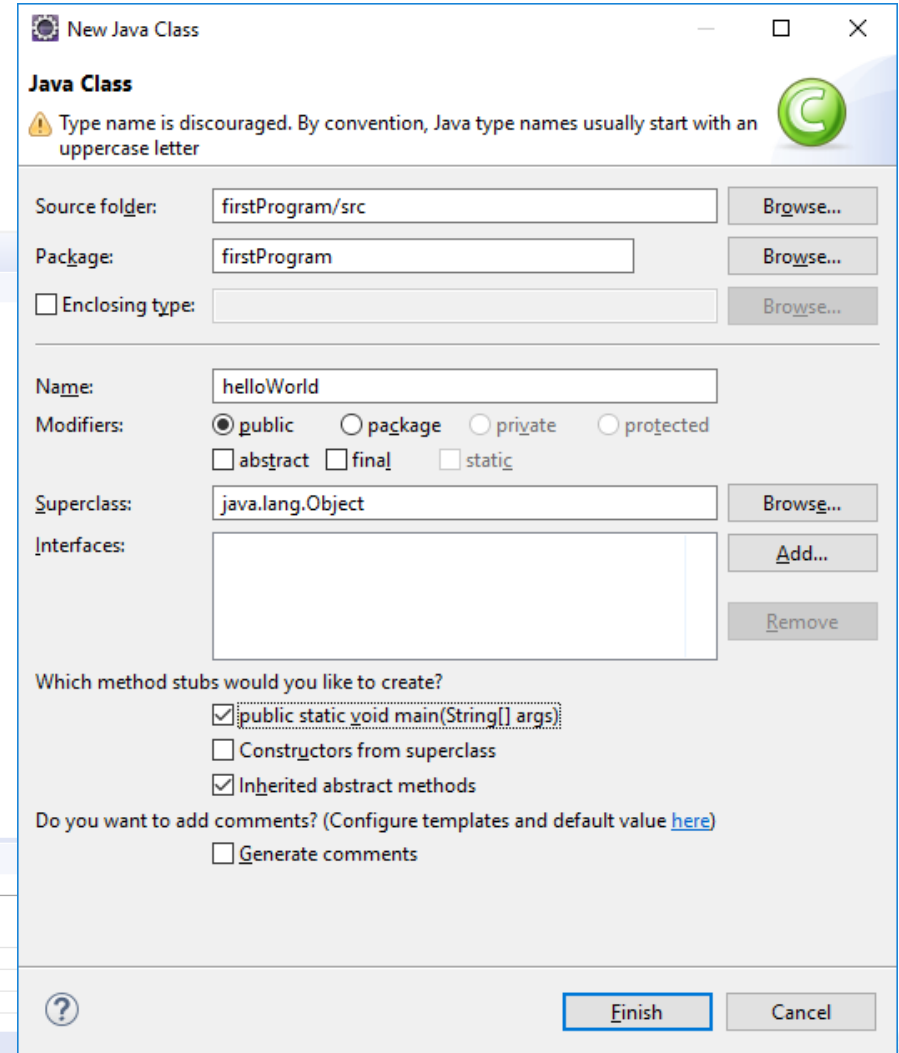
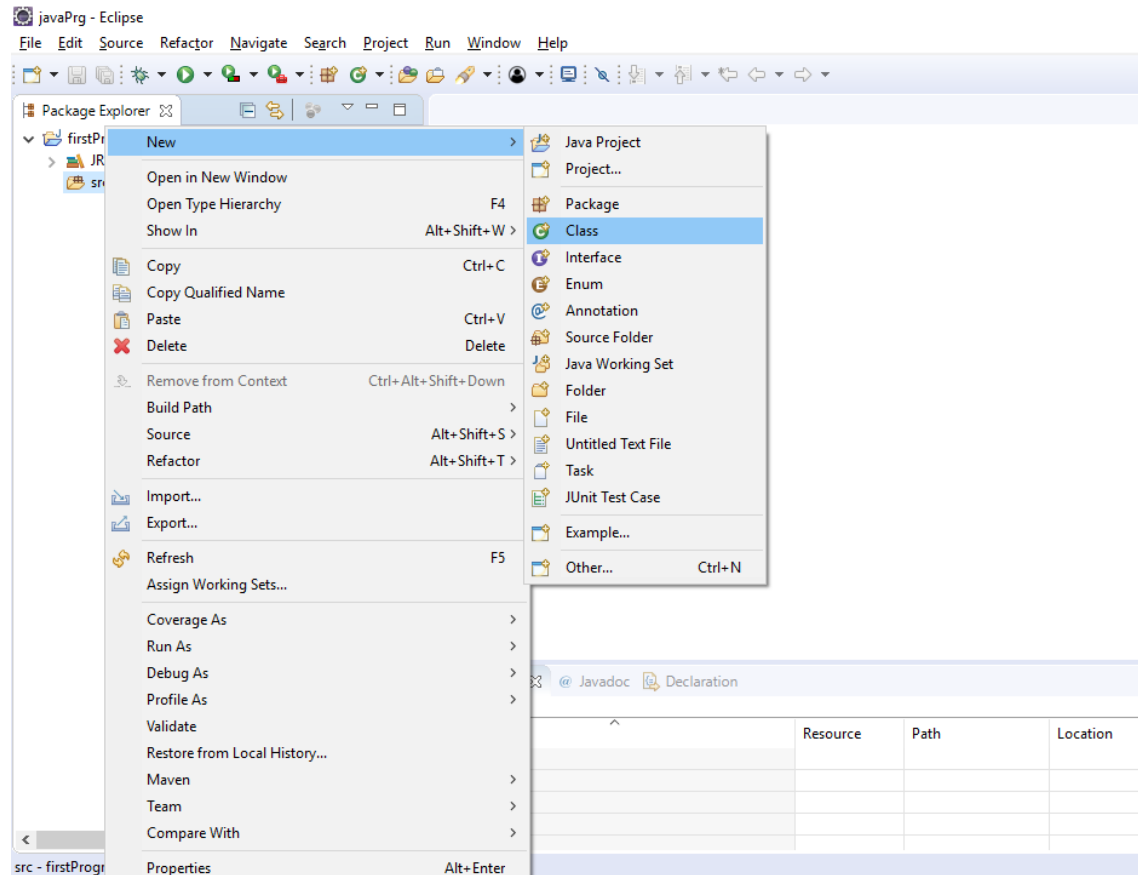


## ■ Package explorer & các thư viện mặc định



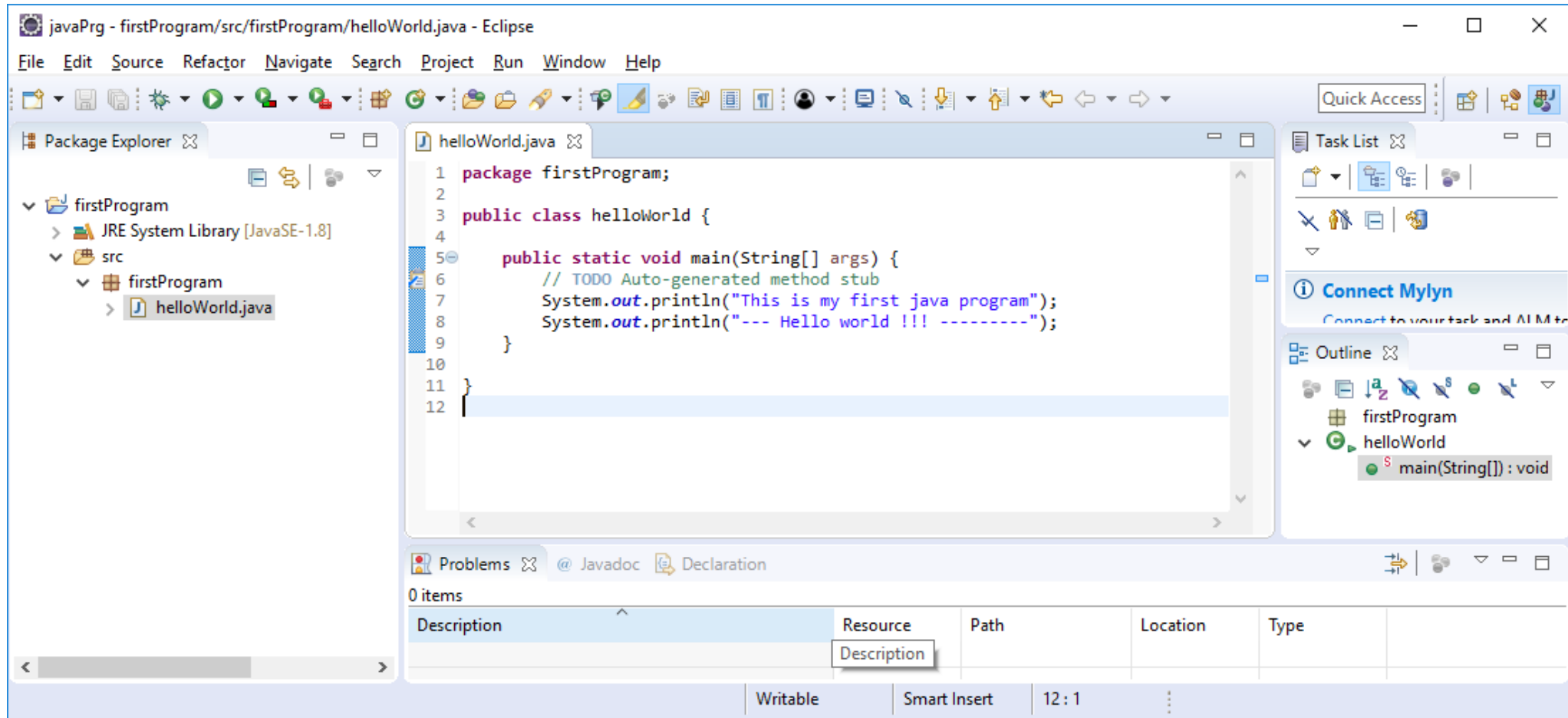
# ■ Tạo class đầu tiên cho chương trình

right mouse -> thư mục src, chọn new -> class

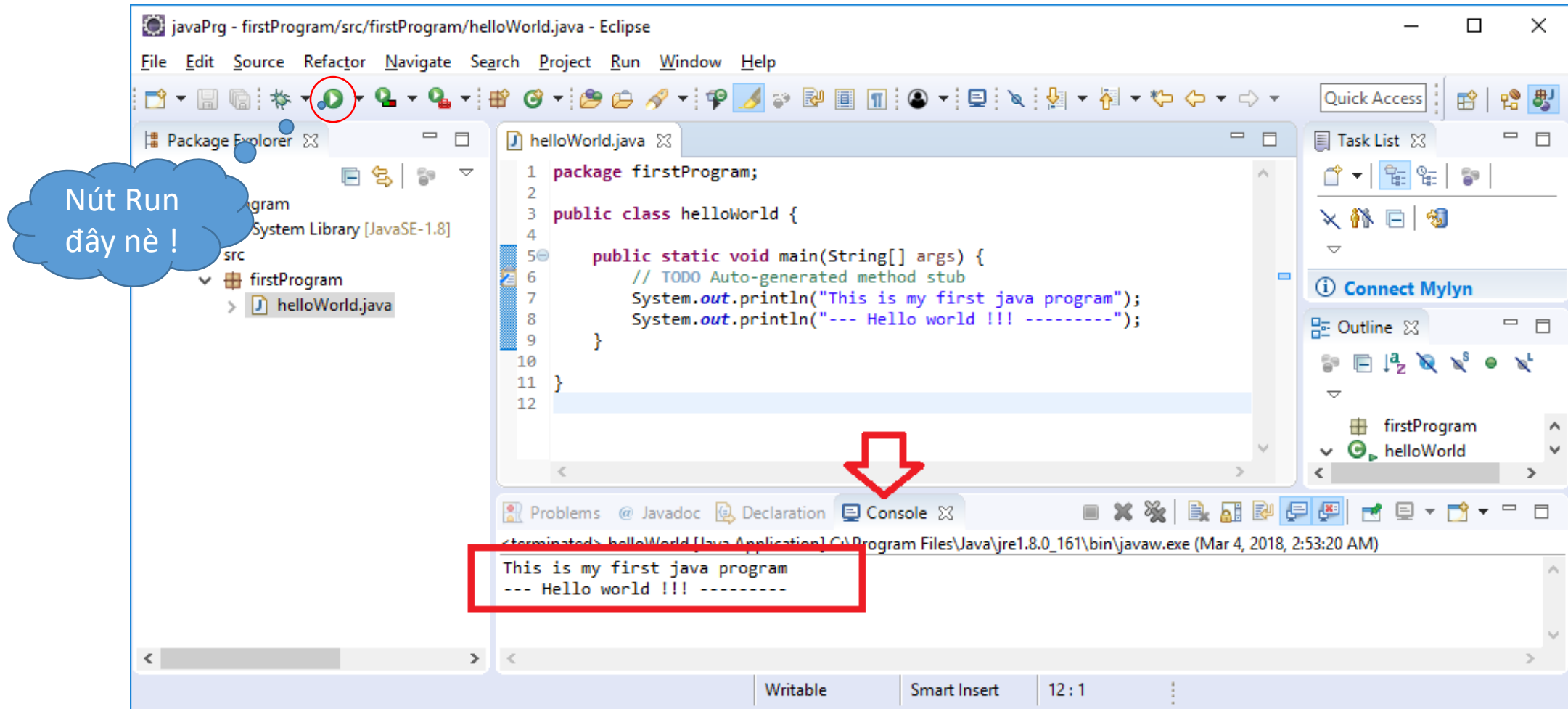




## ■ Xuất dữ liệu với **System.out.println**(“Chuỗi dữ liệu”)



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



# 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.
  - ❖ ...

# Nhớ gì ?!!!

- 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**”, 9<sup>th</sup> edition, 2012
- Oracle, “**The Java™ Tutorials**”,  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>,  
6:20PM, 18/01/2018
- Java tutorial, <https://www.javatpoint.com/java-tutorial> , 6:20PM,  
18/01/2018