



# NGÔN NGỮ LẬP TRÌNH JAVA

**Gv:** Sử Nhật Hạ, M.S.

**Email:** [sunhatha@gmail.com](mailto:sunhatha@gmail.com)

**Đt:** 0907557272

# **Bài 1: Tổng Quan Về Ngôn Ngữ Lập Trình Java**

---

- 1. Lịch sử ra đời ngôn ngữ lập trình Java**
- 2. Đặc điểm ngôn ngữ lập trình Java**
- 3. Các công nghệ của ngôn ngữ lập trình Java**
- 4. Các ứng dụng của ngôn ngữ lập trình Java**
- 5. Java Core API**
- 6. Môi trường làm việc của Java**
- 7. Cài đặt và hướng dẫn sử dụng NetBeads IDE**

# **1. Lịch sử ra đời ngôn ngữ lập trình Java**

---

1.1. Giới thiệu về ngôn ngữ lập trình Java

1.2. Lịch sử phát triển của ngôn ngữ lập trình Java

# 1.1. Giới thiệu về ngôn ngữ lập trình Java

- Java là ngôn ngữ lập trình hướng đối tượng được phát triển bởi Sun Microsystems
- Java là ngôn ngữ vừa biên dịch, vừa thông dịch
  - Đầu tiên, mã nguồn được biên dịch bằng công cụ **JAVAC** để chuyển thành dạng byte code
  - Sau đó, được thực thi trên từng loại máy cụ thể nhờ chương trình thông dịch
- Mục tiêu của các nhà thiết kế Java là cho phép người lập trình viết chương trình một lần nhưng có thể chạy trên bất kỳ phần cứng nào (**Write Once, Run Anywhere**)
- Java được sử dụng rộng rãi để viết chương trình chạy trên internet
- Java là ngôn ngữ lập trình độc lập với thiết bị, không phụ thuộc vào hệ điều hành

# Sun Microsystems



- Công ty Sun Microsystems thành lập năm 1983, là công ty sản xuất phần mềm, bóng bán dẫn máy tính
- Năm 1995, Sun Microsystems đã cho ra đời ngôn ngữ lập trình Java
- Năm 2009, Sun Microsystems chính thức sáp nhập vào Oracle



## 1.2. Lịch sử phát triển của ngôn ngữ lập trình Java

- James Gosling, Mike Sheridan và Patrick Naughton khởi xướng dự án ngôn ngữ Java vào tháng 6 năm 1991
- Ban đầu ngôn ngữ này được gọi là Oak (cây sồi). Sau đó được đổi tên thành Java (tên hòn đảo trồng nhiều cà phê thuộc Indonesia)
- Các phiên bản Java đã phát hành:

Version	Date	Version	Date
JDK Beta	17-Jun-1995	Java SE 8	18-Mar-2014
JDK1.0	23-Jan-1996	Java SE 9	21-Sep-2017
JDK 1.1	19-Feb-1997	Java SE 10	20-Mar-2018
J2SE 1.2	8-Dec-1998	Java SE 11	25-Sep-2018
J2SE 1.3	8-May-2000	Java SE 12	19-Mar-2019
J2SE 1.4	6-Feb-2002	Java SE 13	17-Sep-2019
J2SE 5.0	30-Sep-2004	Java SE 14	17-Mar-2020
Java SE 6	11-Dec-2006	Java SE 15	15-Sep-2020
Java SE 7	28-Jul-2011		

## 2. Đặc điểm ngôn ngữ lập trình Java

Có 7 đặc điểm nổi bật của ngôn ngữ lập trình Java:

1. **Máy ảo Java (JVM – Java Virtual Machine):** Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo Java (Java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã Java bytecode thành mã máy tương ứng
2. **Ngôn ngữ thông dịch:** Java là ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình viết bằng ngôn ngữ lập trình Java có đuôi (.java) được biên dịch thành tập tin có đuôi (.class), sau đó sẽ được trình thông dịch (JVM) dịch thành mã máy tương ứng
3. **Độc lập nền tảng phần cứng và hệ điều hành:** Chương trình viết bằng ngôn ngữ lập trình Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux, ...) thông qua máy ảo Java (JVM)

## 2. Đặc điểm ngôn ngữ lập trình Java ...

Có 7 đặc điểm nổi bật của ngôn ngữ lập trình Java:

4. **Ngôn ngữ lập trình hướng đối tượng:** hướng đối tượng trong Java giống như trong ngôn ngữ lập trình C++. Nhưng trong Java không có đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm **interface** để hỗ trợ đa kế thừa
5. **Đa nhiệm, đa luồng (Multi tasking - Multi threading):** Java hỗ trợ lập trình đa nhiệm, đa luồng, cho phép nhiều tiến trình có thể chạy song song và tương tác với nhau
6. **Cơ chế thu gom rác tự động:** khi tạo ra các đối tượng trong Java, JRE sẽ cấp phát không gian bộ nhớ cho các đối tượng ở trong heap. Bộ thu gom rác tự động của Java sẽ theo vết các tài nguyên đã cấp. Khi không có tham chiếu nào đến vùng nhớ, bộ thu gom rác sẽ tiến hành thu hồi vùng nhớ đã cấp theo định kỳ



## 2. Đặc điểm ngôn ngữ lập trình Java ...

### 7. Tính an toàn và bảo mật:

- ✓ **Tính an toàn:** Ngôn ngữ lập trình Java yêu cầu chắc chắn về kiểu dữ liệu:
  - Dữ liệu phải được khai báo tường minh
  - Không sử dụng con trỏ và các phép toán với con trỏ
  - Java kiểm soát chặt chẽ việc truy xuất mảng, chuỗi, không cho sử dụng kỹ thuật tràn, do đó các truy xuất sẽ không vượt quá kích thước của mảng hoặc chuỗi
  - Quá trình cấp phát và giải phóng bộ nhớ được thực hiện tự động
  - Cơ chế xử lý lỗi giúp việc xử lý và phục hồi dễ dàng

## 2. Đặc điểm ngôn ngữ lập trình Java ...

### 7. Tính an toàn và bảo mật: ...

- ✓ **Tính bảo mật:** Java cung cấp một môi trường quản lý chương trình với nhiều mức khác nhau:
  - **Mức 1:** chỉ có thể truy xuất dữ liệu cũng như phương thức thông qua giao diện mà lớp cung cấp
  - **Mức 2:** trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ lập trình Java trước khi thông dịch
  - **Mức 3:** trình thông dịch sẽ kiểm tra mã byte code xem các đoạn mã này có đảm bảo được các quy định, quy tắc trước khi thực thi
  - **Mức 4:** Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống

# 3. Các công nghệ của ngôn ngữ lập trình Java

## Có 4 nền tảng Java:

1. **Java SE (Java Standard Edition):** bao gồm các API lập trình Java như java.lang, java.io, java.net, java.util, java.sql, java.math, .... Nó bao gồm các chủ đề cốt lõi như OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, ...
2. **Java EE (Java Enterprise Edition):** đây là một nền tảng doanh nghiệp chủ yếu được sử dụng để phát triển các ứng dụng web và doanh nghiệp. Java EE được xây dựng trên nền tảng Java SE. Java EE bao gồm các chủ đề như Servlet, JSP, Web Services, EJB, JPA, ...
3. **Java ME (Java Micro Edition):** đây là một nền tảng vi mô chủ yếu được sử dụng để phát triển các ứng dụng di động
4. **JavaFX:** là một nền tảng phần mềm phát triển các ứng dụng Internet phong phú (Rich Internet Applications – RIAs) có thể chạy trên nhiều loại thiết bị, nhiều hệ điều hành khác nhau. JavaFX là một giải pháp công nghệ cho GUI trên nền tảng Java nhằm tạo giao diện đồ họa người dùng dựa trên Swing và Java2D

# 4. Các ứng dụng của ngôn ngữ lập trình Java

Java đã được sử dụng trong các lĩnh vực khác nhau như:

1. Desktop app
2. Web app
3. Enterprise App
4. Mobile app
5. Game
6. Hệ thống nhúng
7. Smart card
8. Robot
9. ...

## 4.1. Các kiểu của Java app

Có 4 kiểu ứng dụng chính của Java app:

1. **Standalone app:** Standalone App cũng được biết đến như Desktop App hoặc Window-based App. Để tạo ra ứng dụng kiểu này người ta thường sử dụng AWT, Swing hoặc JavaFX framework
2. **Web app:** là ứng dụng chạy trên server và tạo được các trang động. Hiện nay, servlet, jsp, struts, jsf, spring... là những công nghệ được sử dụng để tạo Web App trong java
3. **Enterprise app:** Một ứng dụng dạng như Banking App, có lợi thế là tính bảo mật cao, cân bằng tải (load balancing) và clustering. Trong java, EJB được sử dụng để tạo các Enterprise App
4. **Mobile app:** là ứng dụng được tạo ra cho các thiết bị di động. Hiện nay Android và Java ME được sử dụng để chạy các ứng dụng này

## 5. Java Core API

Java Core API được đưa ra qua phiên bản JFC 1.1. Một số package thường được dùng như sau:

1. **java.applet:** Đây là package nhỏ nhất chứa một mình lớp Applet. Các lớp Applet nhúng trong trang Web đều dẫn xuất từ lớp này
2. **java.awt:** Package này được gọi là Abstract Window Toolkit (AWT). Chúng chứa các tài nguyên dùng để tạo giao diện đồ họa. Một số lớp bên trong là: Button, GridBagLayout, Graphics, ...
3. **java.io:** Cung cấp thư viện nhập xuất chuẩn của ngôn ngữ. Chúng cho phép tạo và quản lý dòng dữ liệu file, inputStream, outputStream, ...
4. **java.util:** Package này cung cấp một số công cụ hữu ích. Một vài lớp của package này là: Date, Hashtable, Stack, Vector và StringTokenizer, ...
5. **java.net:** Cung cấp khả năng giao tiếp với máy từ xa. Cho phép tạo và kết nối với Socket hoặc URL

## 5. Java Core API ...

6. **java.awt.event**: Chứa các lớp dùng để xử lý các sự kiện trong chương trình như chuột, bàn phím
7. **java.rmi**: Công cụ để gọi hàm từ xa. Chúng cho phép tạo đối tượng trên máy khác và sử dụng các đối tượng đó trên máy cục bộ
8. **java.security**: Cung cấp các công cụ cần thiết để mã hóa và đảm bảo tính an toàn của dữ liệu truyền giữa máy trạm và máy chủ
9. **java.sql**: Package này chứa Java DataBase Connectivity (JDBC), dùng để truy xuất cơ sở dữ liệu quan hệ như Oracle, SQL Server

## 6. Môi trường làm việc của Java

---

1. JDK - Java Development Kit
2. JVM - Java Virtual Machine
3. Quy trình biên dịch và thông dịch của java
4. Kiểu biên dịch và thực thi của Java so với C++



## 6.1. JDK - Java Development Kit

JDK còn được gọi là SDK (Java SDK - Software Development Kit) là bộ công cụ phát triển ứng dụng Java bao gồm 4 thành phần: Classes, Compiler, Debugger, Java Runtime Environment

Bộ công cụ này bao gồm:

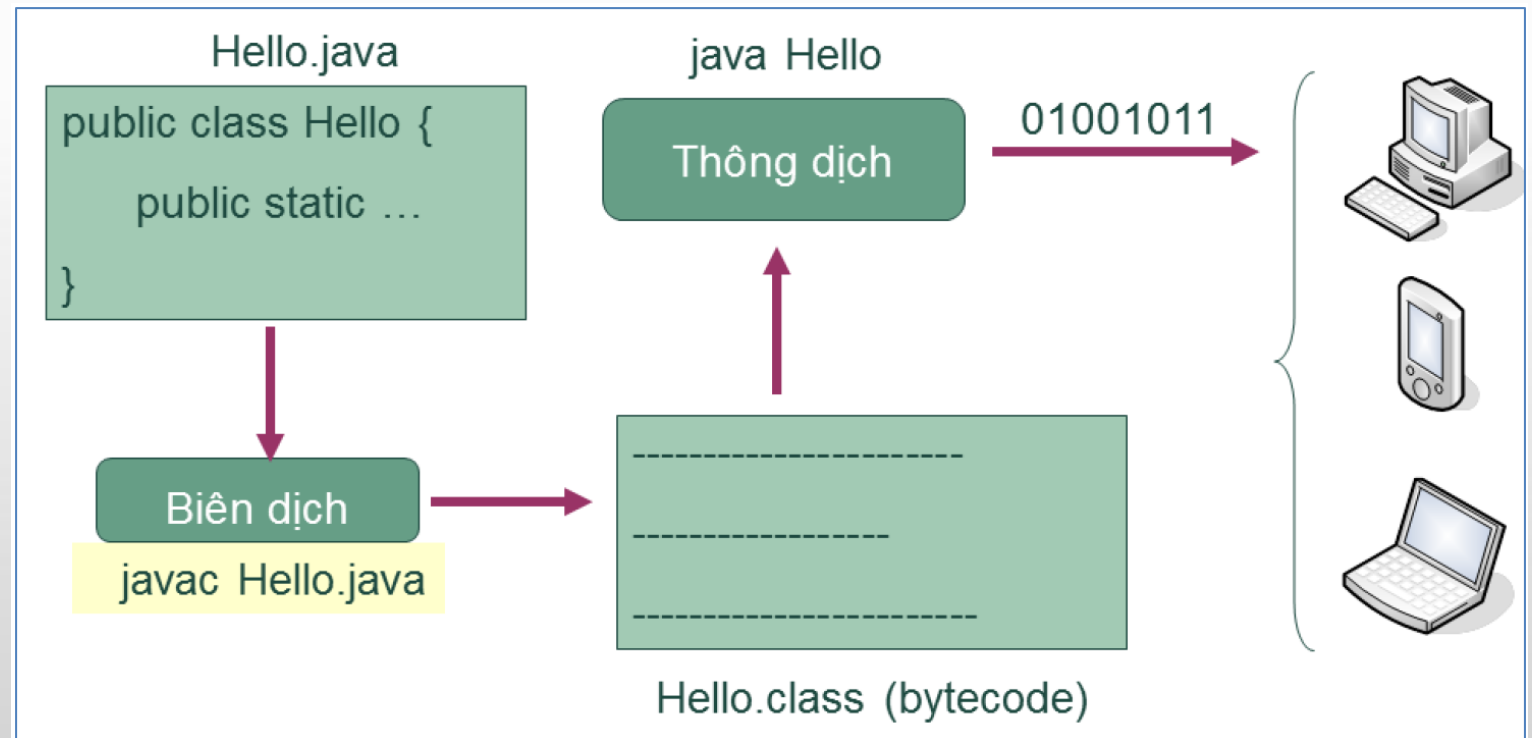
1. **Javac:** chương trình dịch mã nguồn sang bytecode
2. **Java:** bộ thông dịch thực thi java application
3. **Appletviewer:** bộ thông dịch thực thi java applet mà không cần sử dụng trình duyệt
4. **Javadoc:** bộ tạo tài liệu dạng HTML từ mã nguồn và chú thích
5. **Jdb:** bộ gỡ lỗi (java debugger)
6. **Javap:** trình dịch ngược bytecode
7. **Jar:** dùng để đóng gói lưu trữ các module viết bằng Java (tạo ra file **.jar**), là phương pháp tiện lợi để phân phối những chương trình Java

## 6.2. JVM - Java Virtual Machine

- JVM là một phần mềm dựa trên cơ sở máy tính ảo. JVM cũng được xem như là một hệ điều hành thu nhỏ.
- Máy ảo phụ thuộc vào Platform (phần cứng, OS), nó cung cấp môi trường thực thi cho Java (độc lập với platform).
- JVM thiết lập cho các mã Java đã biên dịch có một cái nhìn trong suốt (trasparence) về các phần cứng bên dưới

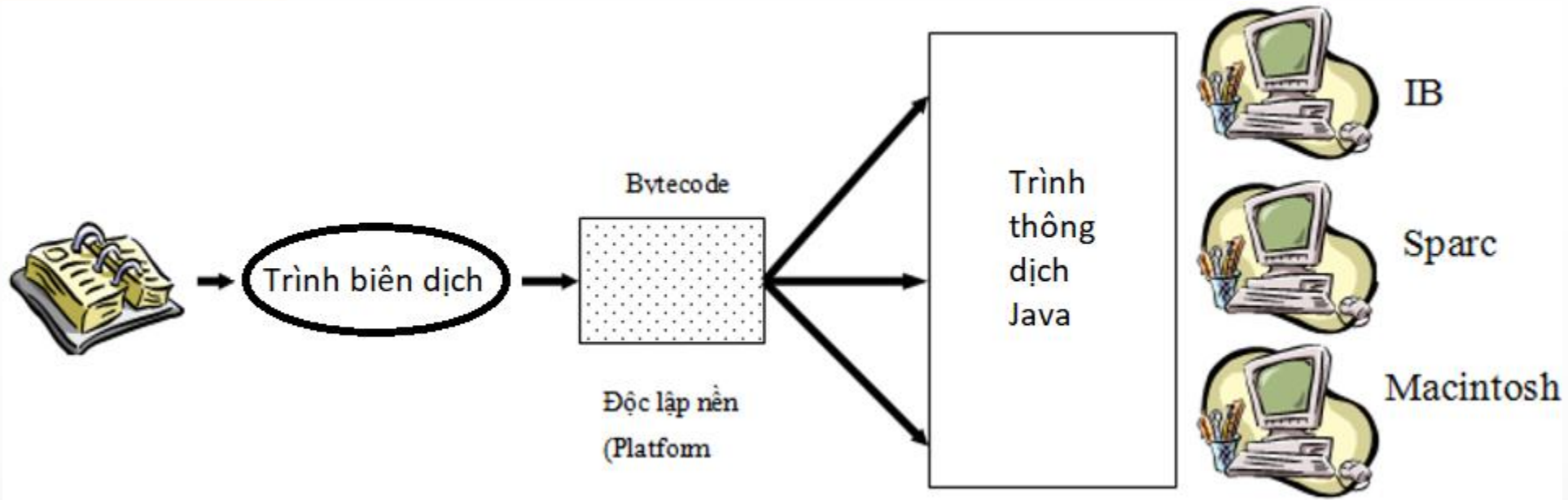
## 6.3. Quy trình biên dịch và thông dịch của java

- Trình biên dịch chuyển mã nguồn thành tập các lệnh (byte code) không phụ thuộc vào phần cứng và hệ điều hành cụ thể
- Trình thông dịch trên mỗi máy chuyển tập lệnh này thành chương trình thực thi
- Máy ảo tạo ra một môi trường để thực thi các lệnh bằng cách:
  1. Nạp các file .class
  2. Quản lý bộ nhớ
  3. Thu gom rác

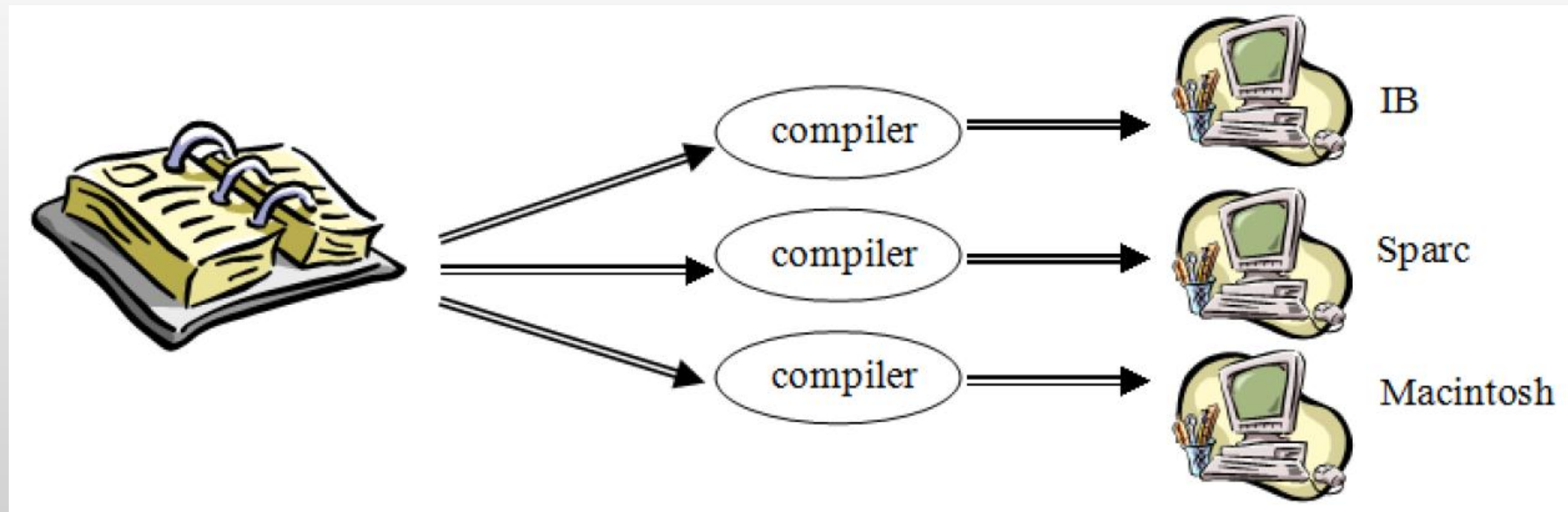


## 6.4. Kiểu biên dịch và thực thi của Java so với C++

- Java



- C++



# 7. Cài đặt và hướng dẫn sử dụng NetBeads IDE

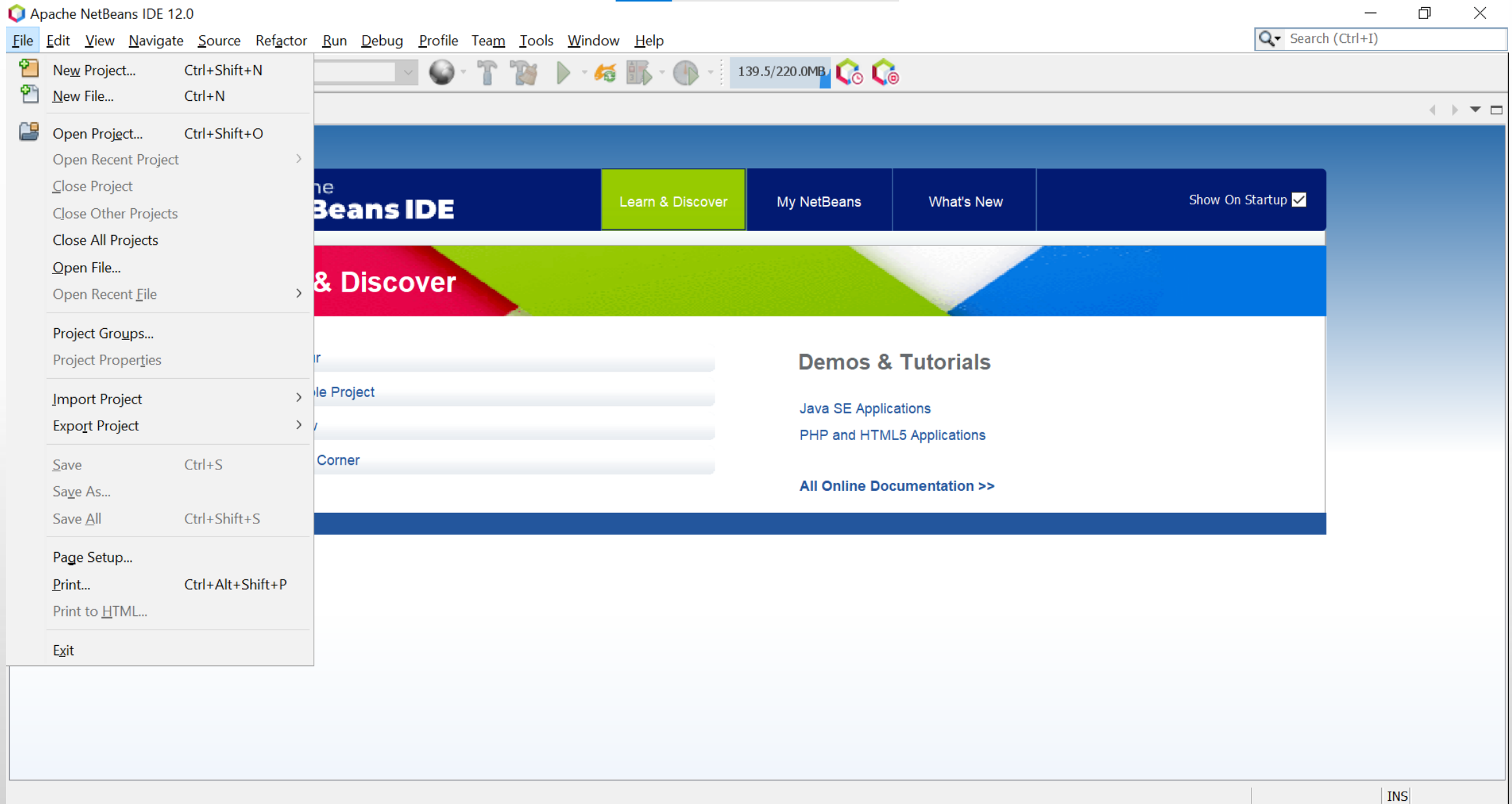
Các bộ công cụ phát triển ứng dụng Java tiêu biểu:

1. Jcreator
2. NetBeans
3. Eclipse
4. EditPlus
5. Borland Jbuilder
6. Java Studio của Sun
7. ...

## 7.1 Giới thiệu NetBeans IDE

- NetBeans IDE Là môi trường phát triển - một công cụ dành cho lập trình viên để viết code, biên dịch, gỡ lỗi (debug) và triển khai (deploy) chương trình được viết bằng Java
- Netbeans IDE với giao diện thân thiện, phổ biến, miễn phí và hiệu quả là lựa chọn tối ưu nhất để viết Java
- NetBeans IDE dễ cài đặt và chạy trên nhiều hệ điều hành: Windows, Linux, Mac OS X và Solaris

# Màn hình làm việc của Apache NetBeans IDE



# Cài đặt Apache NetBeans

---

1. Download Java SDK:

<https://www.oracle.com/java/technologies/javase-downloads.html>

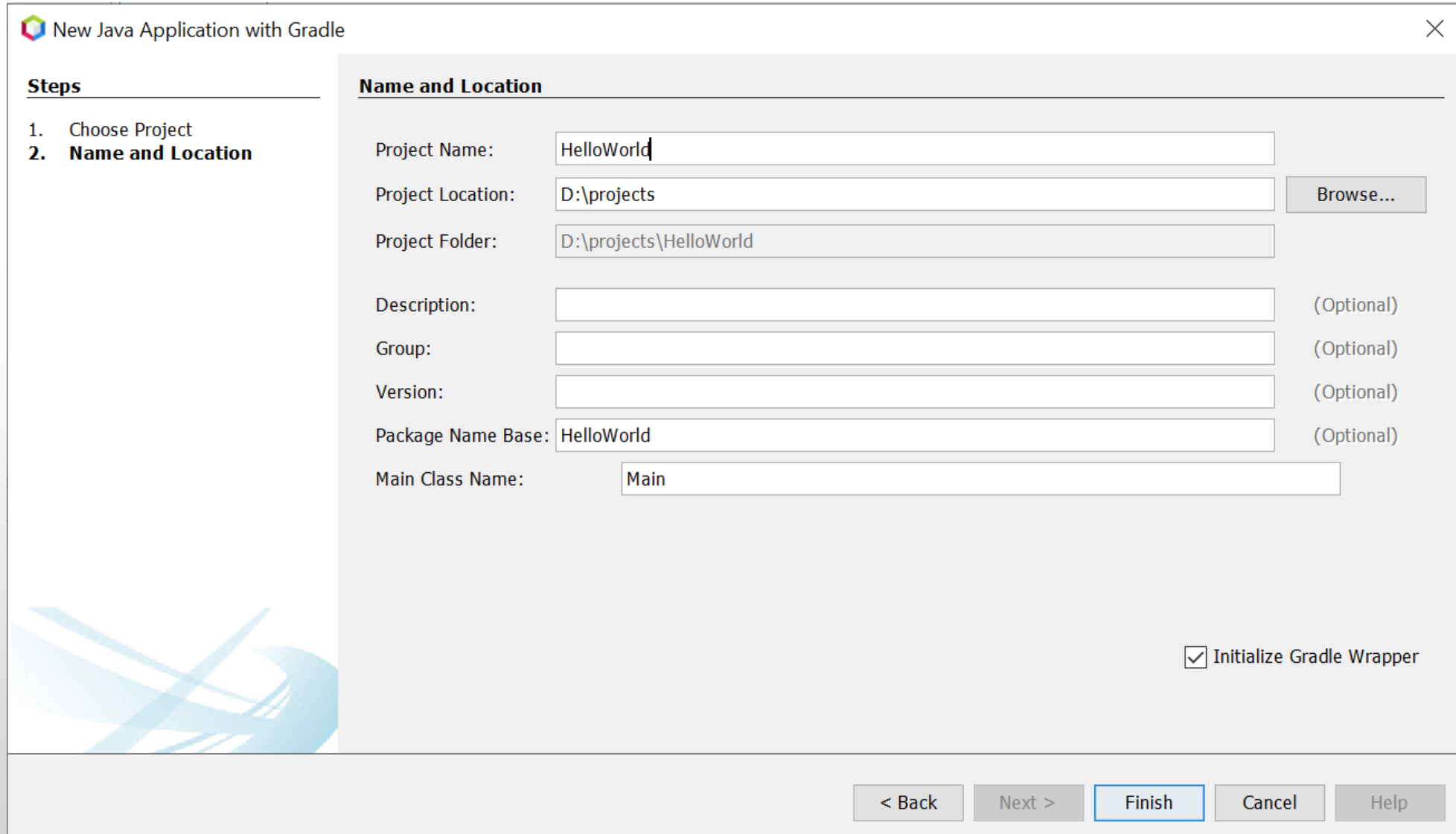
2. Download Apache NetBeans:

<https://netbeans.apache.org/download/index.html>



# Hướng dẫn sử dụng Apache NetBeans

- Mở Apache Netbeans sau đó chọn File -> New Project



New Java Application with Gradle

**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name: HelloWorld

Project Location: D:\projects Browse...

Project Folder: D:\projects\HelloWorld

Description: (Optional)

Group: (Optional)

Version: (Optional)

Package Name Base: HelloWorld (Optional)

Main Class Name: Main

☒ Initialize Gradle Wrapper

< Back Next > Finish Cancel Help

# Màn hình soạn thảo cho ứng dụng

HelloWorld - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

225.2/332.0MB

Projects x Files Services

- HelloWorld
  - Source Packages [java]
    - HelloWorld
      - Main.java
  - Source Packages [generated]
  - Configurations
  - Build Scripts
  - Java Dependencies

main - Navigator x

Members <empty>

- Main
  - Main()
  - main(String[] args)

Start Page x Main.java x

Source History

```
4  * and open the template in the editor.
5  */
6  package HelloWorld;
7
8  /**
9   *
10  * @author hasu
11  */
12  public class Main {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          System.out.println("Hello World!");
20      }
21
22  }
```

HelloWorld.Main > main >

Output - Run (HelloWorld) x

```
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :run
Hello World!

BUILD SUCCESSFUL in 262ms
2 actionable tasks: 1 executed, 1 up-to-date
```

# Một số phím tắt thông dụng của Apache NetBeans

## • Tìm và thay thế

- **Ctrl-F3:** Tìm từ tại vị trí con trỏ
- **F3/Shift-F3:** Tìm từ kế tiếp / trước đó trong file
- **Ctrl-F/H:** Tìm/thay thế từ trong file
- **Alt-F7:** Tìm vị trí hàm được sử dụng trong project
- **Ctrl-R:** Đổi tên biến
- **Alt-Shift-L:** Copy đường dẫn file đang mở

## • Coding trong Java

- **Alt-Insert:** Sinh code
- **Ctrl-Shift-I:** Tự động import class cần thiết
- **Alt-Shift-F:** Định dạng code
- **Ctrl-Shift-Up/D:** Copy dòng tại vị trí con trỏ
- **Ctrl-/:** Bật/tắt comment
- **Ctrl-E/X:** Xóa dòng tại vị trí con trỏ

# Một số phím tắt thông dụng của NetBeans...

- **Biên dịch, kiểm tra và chạy project**

- **F9:** Compile file
- **F11:** Build project
- **Shift-F11:** Clean và build project
- **Shift-F6:** Run file
- **F6:** Run project

- **Debugging**

- |  |                            |
|--|----------------------------|
| • <b>Ctrl-F5:</b> Debug project            | • <b>F4:</b> Run to cursor |
| • <b>Ctrl-Shift-F5:</b> Debug file         | • <b>F5:</b> Continue      |
| • <b>Shift-F5:</b> Finish debugger session | • <b>F7:</b> Step into     |
| • <b>Ctrl-F8:</b> Toggle line breakpoint   | • <b>Ctrl-F7:</b> Step out |
| • <b>Ctrl-Shift-F8:</b> New breakpoint     | • <b>F8:</b> Step over     |

# Cấu trúc chương trình Java cơ bản

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package HelloWorld;
7
8  /**
9   *
10   * @author hasu
11   */
12  public class Main {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          System.out.println("Hello World!");
20      }
21
22  }
```