

BÀI 1: TỔNG QUAN VỀ NGÔN NGỮ JAVA

Học xong bài này người học sẽ:

- ✓ Trình bày được định nghĩa về ngôn ngữ lập trình Java, lịch sử ra đời và phát triển
- ✓ Trình bày được các đặc trưng của ngôn ngữ lập trình Java
- ✓ Trình bày được các công nghệ và các kiểu ứng dụng từ ngôn ngữ lập trình Java
- ✓ Cài đặt được môi trường làm việc cho ngôn ngữ lập trình Java
- ✓ Lập trình được các chương trình Java đơn giản thao tác trên các kiểu dữ liệu cơ sở, các cấu trúc lập trình điều khiển, ... trên môi trường Netbeans..

1.1 GIỚI THIỆU VỀ NGÔN NGỮ LẬP TRÌNH JAVA

1.1.1 Giới thiệu về ngôn ngữ lập trình Java

Java là ngôn ngữ lập trình được Sun Microsystems giới thiệu vào tháng 6 năm 1995. Là ngôn ngữ lập trình hướng đối tượng. Java có thể giải quyết hầu hết các công việc mà các ngôn ngữ khác có thể làm được.

Java là ngôn ngữ vừa biên dịch vừa thông dịch. Đầu tiên mã nguồn được 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 kì phần cứng nào.

Ngày nay, Java được sử dụng rộng rãi để viết chương trình chạy trên Internet. Là ngôn ngữ lập trình hướng đối tượng độc lập thiết bị, không phụ thuộc vào hệ điều hành. Java không chỉ dùng để viết các ứng dụng chạy đơn lẻ hay trong mạng mà còn để xây dựng các trình điều khiển thiết bị cho điện thoại di động, PDA

Ngôn ngữ lập trình Java đã trở thành một công cụ lập trình của các lập trình viên chuyên nghiệp. Java được xây dựng trên nền tảng của C và C++. Do vậy nó sử dụng các cú pháp của C và các đặc trưng hướng đối tượng của C++.



Công ty Sun Microsystems thành lập năm 1983 là công ty sản xuất phần mềm, bóng bán dẫn và máy tính. Năm 1995 Sun đã cho ra đời ngôn ngữ lập trình Java. Năm 2009 Sun chính thức sáp nhập về Google, kết thúc 30 hình thành phát triển và thịnh vượng, chuyển sang giai đoạn mới của công nghệ.

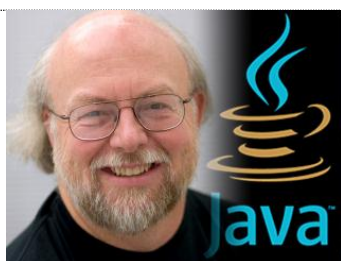
1.1.2 Lịch sử phát triển của ngôn ngữ lập trình Java

Vào năm 1991, một nhóm các kỹ sư của Sun Microsystems có ý định thiết kế một ngôn ngữ lập trình để điều khiển các thiết bị điện tử như tivi, máy giặt, lò nướng ... Mặc dù C và

C++ có khả năng làm được việc này nhưng trình biên dịch lại phụ thuộc vào từng loại CPU. Trình biên dịch thường phải tốn nhiều thời gian để xây dựng nên rất đắt. Vì vậy để mỗi loại CPU có một trình biên dịch riêng là rất tốn kém.

Do đó nhu cầu thực tế đòi hỏi một ngôn ngữ chạy nhanh, gọn, hiệu quả và độc lập thiết bị tức là có thể chạy trên nhiều loại CPU khác nhau, dưới các môi trường khác nhau. “Oak” (cây sồi) đã ra đời vào năm 1995, sau đó được đổi tên thành Ngôn ngữ lập trình Java (tên hòn đảo có trồng nhiều Cà phê).

Mục đích của Java để phát triển ứng dụng cho các thiết bị điện tử thông minh. Hiện nay Java được sử dụng để phát triển nhiều loại ứng dụng khác nhau như: Cơ sở dữ liệu, mạng, Internet, games, viễn thông,...



Cuối 1990, James Gosling và các cộng sự, được Cty Sun Microsystems giao nhiệm vụ xây dựng phần mềm lập trình cho các mặt hàng điện tử để cài chương trình vào các bộ xử lý của các thiết bị.

1.2 CÁC ĐẶC TRƯNG CỦA NGÔN NGỮ JAVA

1.2.1 Đơn giản

Do những người thiết kế mong muốn phát triển một ngôn ngữ dễ đọc và quen thuộc với đa số người lập trình, nên Java được loại bỏ các đặc trưng phức tạp của C và C++ như thao tác con trỏ, thao tác nạp chồng (overload) ... và sử dụng lệnh “goto” cũng như file header (.h). Cấu trúc “struct” và “union” cũng được loại bỏ khỏi Java.

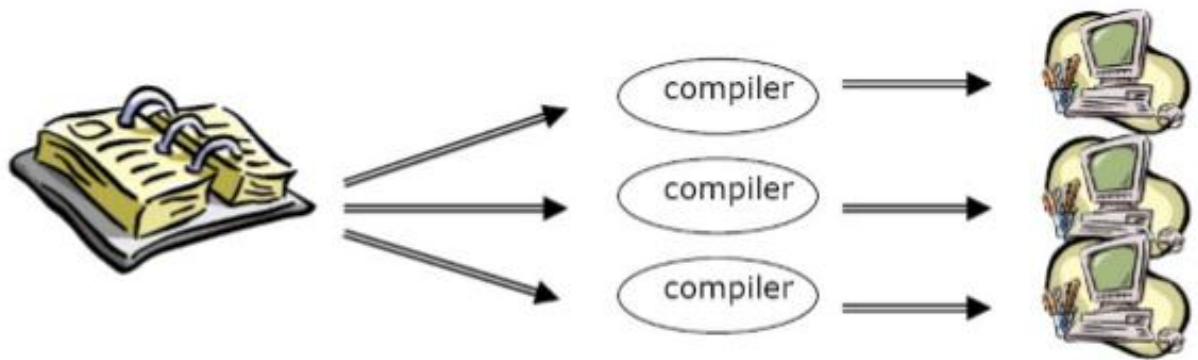
1.2.2 Hướng đối tượng

Java được thiết kế quanh mô hình hướng đối tượng. Vì vậy trong Java, tiêu điểm là dữ liệu và các phương pháp thao tác lên dữ liệu đó.

1.2.3 Dễ bảo trì (Vận hành)

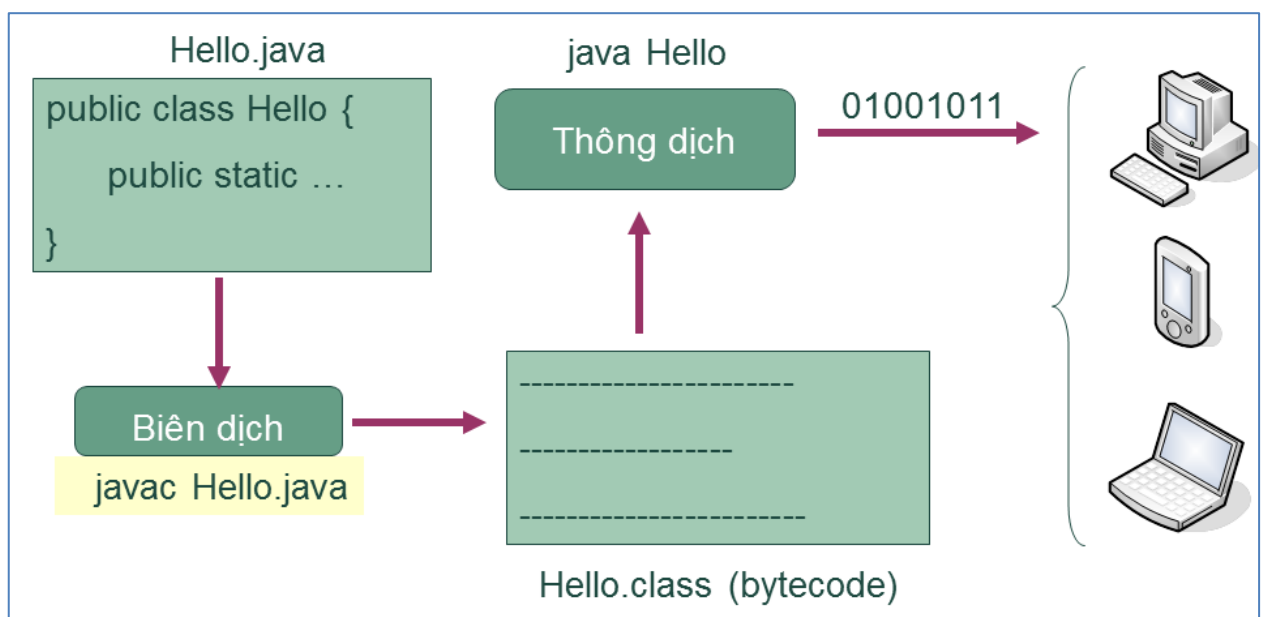
Đây là khả năng một chương trình được viết tại một máy nhưng có thể chạy được bất kỳ đâu. Chúng được thể hiện ở mức mã nguồn và mức nhị phân: Ở mức mã nguồn, người lập trình cần mô tả kiểu cho mỗi biến. Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một thư viện các lớp cơ sở. Vì vậy chương trình Java được viết trên một máy có thể được dịch và chạy trên các loại máy khác mà không cần viết lại.

Ở mức nhị phân, một chương trình đã biên dịch có thể chạy trên nền khác mà không cần dịch lại mã nguồn. Tuy vậy cần có phần mềm máy ảo Ngôn ngữ lập trình Java , hoạt động như một trình thông dịch tại máy thực thi.



Hình 1.1: Minh họa biên dịch và chạy chương trình C/C++.

Trình biên dịch sẽ chuyển các chương trình viết bằng C/C++ hay ngôn ngữ khác thành mã máy nhưng phụ thuộc vào CPU. Nên khi muốn chạy trên CPU khác, ta phải biên dịch lại chương trình (hình 1.1).



Hình 1.2: Minh họa biên dịch và chạy chương trình Ngôn ngữ lập trình Java

Môi trường phát triển của Java được chia làm hai phần: Trình biên dịch và trình thông dịch. Không như C hay C++, trình biên dịch của Java chuyển mã nguồn thành dạng byte code độc lập với phần cứng mà có thể chạy trên bất kỳ CPU nào.

Nhưng để thực thi chương trình dưới dạng byte code, tại mỗi máy cần phải có trình thông dịch của Java hay còn gọi là máy ảo Java. Máy ảo Java chuyển byte code thành mã lệnh mà CPU thực thi được.

1.2.4 Mạnh mẽ

Java yêu cầu chặt chẽ về kiểu dữ liệu và phải mô tả rõ ràng khi viết chương trình. Chúng sẽ kiểm tra lúc biên dịch và cả trong thời gian thông dịch, vì vậy Java loại bỏ các kiểu dữ liệu dễ gây lỗi.

1.2.5 Bảo mật

Java cung cấp một số lớp để kiểm tra bảo mật:

Ở lớp đầu tiên, dữ liệu và các phương pháp được đóng gói bên trong lớp. Chúng chỉ truy xuất thông qua các giao diện mà lớp cung cấp. Nó cũng ngăn chặn không cho truy xuất thông tin bên ngoài của mảng bằng kỹ thuật tràn và cũng cung cấp kỹ thuật dọn rác trong bộ nhớ. Các đặc trưng này tạo cho Java sự an toàn và khả năng cơ động cao.

Trong lớp thứ hai, trình biên dịch kiểm soát để đảm bảo mã an toàn.

Lớp thứ ba được đảm bảo bởi trình thông dịch. Chúng kiểm tra xem liệu byte code có đảm bảo các quy tắc an toàn trước khi thực thi.

Lớp thứ tư kiểm soát việc nạp các lớp lên bộ nhớ giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

1.2.6 Phân tán

Java có thể được dùng để xây dựng làm việc trên nhiều phần cứng, hệ điều hành và giao diện đồ họa. Java được thiết kế cho các ứng dụng chạy trên mạng. Vì vậy chúng được sử dụng rộng rãi trên Internet, nơi sử dụng nhiều nền tảng khác nhau.

1.2.7 Đa luồng

Chương trình Java sử dụng kỹ thuật đa luồng (Multithread) để thực thi các công việc đồng thời. Chúng cũng cung cấp giải pháp đồng bộ giữa các tiến trình. Đặc tính hỗ trợ đa tiến trình này cho phép xây dựng các ứng dụng trên mạng chạy uyển chuyển.

1.2.8 Động

Java được thiết kế như một ngôn ngữ động để đáp ứng cho những môi trường thực thi mở. các chương trình Java bổ sung thông tin cho các đối tượng tại thời gian thực thi. Điều này cho phép khả năng liên kết động các mã.

1.3 CÁC KIỂU CHƯƠNG TRÌNH CỦA JAVA

-
- Ứng dụng độc lập (console application)
 - Ứng dụng Applets
 - Ứng dụng giao diện (GUI application)
 - Ứng dụng Web (Web application)
 - Ứng dụng nhúng

- Ứng dụng cơ sở dữ liệu
- Ứng dụng Game

1.3.1 Ứng dụng độc lập (console application)

Là ứng dụng nhập xuất ở chế độ văn bản tương tự như màn hình Console của hệ điều hành MS-DOS. Loại chương trình ứng dụng này thích hợp với những ai bước đầu làm quen với ngôn ngữ lập trình java.

Các ứng dụng kiểu Console thường được dùng để minh họa các ví dụ cơ bản liên quan đến cú pháp ngôn ngữ, các thuật toán, và các chương trình ứng dụng không cần thiết đến giao diện người dùng đồ họa.

1.3.2 Ứng dụng Applets

Là loại ứng dụng có thể nhúng và chạy trong trang web. Khi internet mới ra đời, Java Applet cung cấp một khả năng lập trình mạnh mẽ cho các trang web.

Gần đây khi các trình duyệt web đã phát triển với khả năng lập trình bằng VB Script, Java Script, HTML, DHTML, XML,... cùng với sự cạnh tranh khốc liệt giữa Microsoft và Sun đã làm cho Java Applet lu mờ. Đến bây giờ gần như các lập trình viên đều không còn “mặn mà” với Java Applet.

1.3.3 Ứng dụng giao diện (GUI application)

Các chương trình ứng dụng có giao diện người dùng đồ họa trực quan như những chương trình được viết dùng ngôn ngữ lập trình VC++/Visual Basic/C# đã được java giải quyết bằng thư viện AWT và JFC.

JFC (Swing) là thư viện rất phong phú và hỗ trợ mạnh mẽ hơn nhiều so với AWT. JFC giúp cho người lập trình có thể tạo ra một giao diện trực quan của bất kỳ ứng dụng nào.

1.3.4 Ứng dụng Web (Web application)

Java hỗ trợ mạnh mẽ đối với việc phát triển các ứng dụng Web thông qua công nghệ J2EE (Java 2 Enterprise Edition).

Công nghệ J2EE hoàn toàn có thể tạo ra các ứng dụng Web một cách hiệu quả không thua kém công nghệ .NET mà Microsoft.

Công nghệ viết web hiện có của Java là Servlet và Jsp, ngoài ra còn có sự hỗ trợ của lập trình Socket, Java Bean, RMI và CORBA, EJB.

1.3.5 Ứng dụng nhúng

Java Sun đưa ra công nghệ J2ME (The Java 2 Platform, Micro Edition J2ME) hỗ trợ phát triển các chương trình, phần mềm nhúng.

J2ME cung cấp một môi trường cho những chương trình ứng dụng có thể chạy được trên các thiết bị cá nhân như: điện thoại di động (MIDlet), máy tính bỏ túi PDA hay Palm, cũng như các thiết bị nhúng khác.

1.3.6 Ứng dụng cơ sở dữ liệu

Các ứng dụng này sử dụng JDBC API để kết nối tới cơ sở dữ liệu. Java hỗ trợ lập trình kết nối và tương tác được với hầu hết các hệ quản trị CSDL. Như Oracle, SQL Server, MS-Access, MySQL,...

1.3.7 Ứng dụng Game

Lập trình Games bằng Java được phát triển mạnh mẽ. Dùng Java có thể viết được games cho máy desktop và các thiết bị di động.

1.4 CÁC CÔNG NGHỆ JAVA

1.4.1 Java là một công nghệ

Công nghệ Java phát triển mạnh mẽ nhờ vào Sun Microsystem cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại ứng dụng khác nhau.

Java bao gồm:

- Ngôn ngữ lập trình
- Môi trường phát triển
- Môi trường thực thi và triển khai

1.4.2 Các công nghệ Java

- **Desktop applications - J2SE:** Phiên bản chuẩn – Java 2 Standard Edition. J2SE hỗ trợ viết các ứng dụng đơn, ứng dụng client-server.
 - Java Applications: ứng dụng Java thông thường trên desktop
 - Java Applets: ứng dụng nhúng hoạt động trong trình duyệt web
- **Server applications - J2EE:** Nền tảng Java 2, phiên bản doanh nghiệp - Java 2 Enterprise Edition. Hỗ trợ phát triển các ứng dụng thương mại. Chạy trên máy chủ lớn với sức mạnh xử lý và dung lượng bộ nhớ lớn, hỗ trợ gắn liền với servlet, jsp và XML.
- **Mobile (embedded) applications – J2ME:** Phiên bản thu nhỏ - Java 2 Micro Edition. Hỗ trợ viết các ứng dụng trên các thiết bị di động, không dây, thiết bị nhúng,...

1.5 MÔI TRƯỜNG LÀM VIỆC

1.5.1 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:

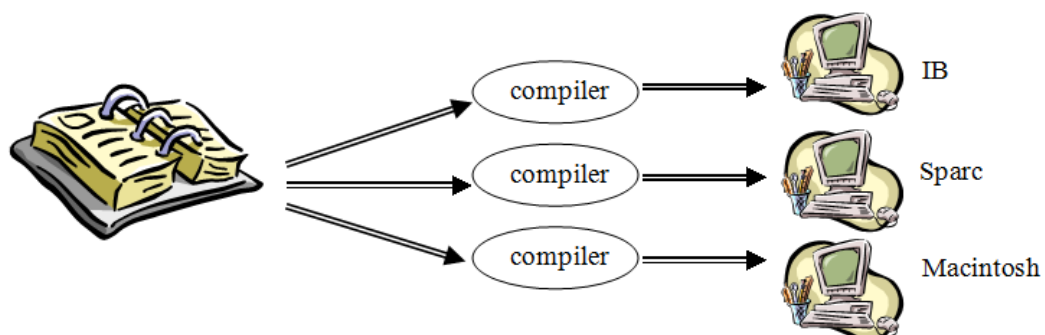
- *Javac*: Chương trình dịch mã nguồn sang *bytecode*
- *Java*: Bộ thông dịch thực thi java application
- *Appletviewer*: Bộ thông dịch thực thi java applet mà không cần sử dụng trình duyệt.
- *Javadoc*: Bộ tạo tài liệu dạng HTML từ mã nguồn và chú thích
- *jdb* Bộ gỡ lỗi (*java debugger*)
- *javap* Trình dịch ngược bytecode
- *jar* Dùng để đóng gói lưu trữ các module viết bằng Java (tạo ra file đuôi .jar), là phương pháp tiện lợi để phân phối những chương trình Java.

Các phiên bản của JDK :

- JDK alpha and Beta (1995)
- JDK 1.0 (23 – 1 – 1996)
- JDK 1.1 (19 – 2 – 1997)
- J2SE 1.2 (8 – 12 – 1998)
- J2SE 1.3 (8 – 5 – 2000)
- J2SE 1.4 (6 – 2 – 2002)
- J2SE 5.0 (30 – 9 – 2004)
- Java SE 6 (11 – 12 – 2006)
- Java SE 7 (28 – 7 – 2011)
- Java SE 8 (18 – 3- 2014)

1.5.2 JVM – Java Virtual Machine

Kiểu dịch của các trình biên dịch ngôn ngữ C/C++:

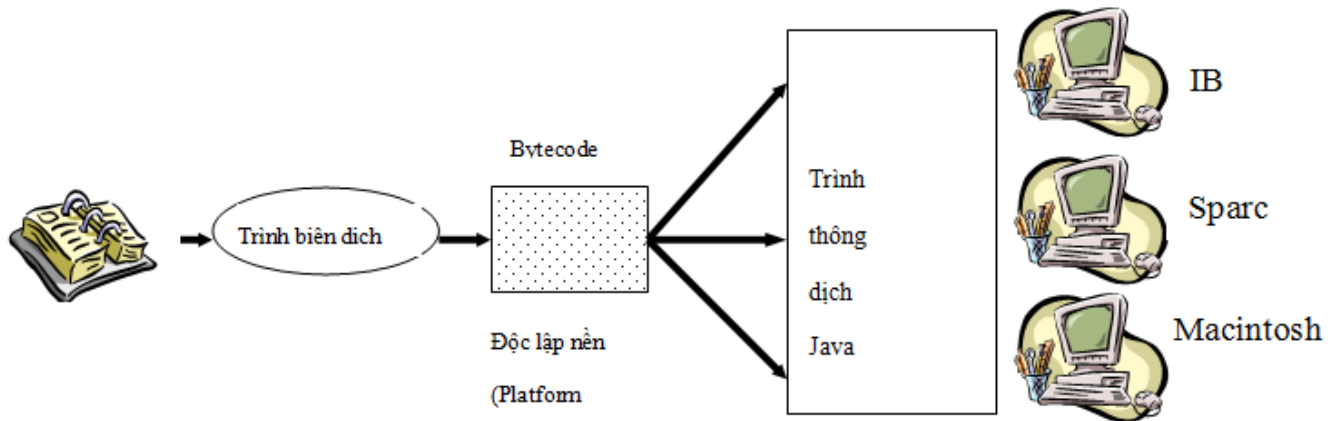


Hình 1.3: Minh họa biên dịch và chạy chương trình C/C++.

Quy trình biên dịch và thông dịch của java:

- Trình biên dịch chuyển mã nguồn thành tập các lệnh không phụ thuộc vào phần cứng 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:
 - Nạp các file .class
 - Quản lý bộ nhớ
 - Dọn “rác”



Hình 1.4: Minh họa biên dịch và chạy chương trình Java.

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).

Nó thiết lập cho các mã Java đã biên dịch có một cái nhìn trong suốt (trasparence) về các phần cứng bên dưới.

1.5.3 Java Core API

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

- 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
- 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.
- java.io: Cung cấp thư viện nhập xuất chuẩn của ngôn ngữ. Chúng cho phép tạo và quản lý dòng dữ liệu theo một vài cách.
- 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.
- java.net: Cung cấp khả năng giao tiếp với máy từ xa. Cho phép tạo và kết nối với Socket hoặc URL.

- java.awt.event: Chứa các lớp dùng để xử lý các sự kiện trong chương trình như chuột, bàn phím.
- 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.
- 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.
- java.sql: Package này chứa Java DataBase Connectivity (JDBC), dùng để truy xuất cơ sở dữ liệu quan hệ như Oracle, SQL Server.

1.5.4 IDE – Integrated Development Environment

1.5.4.1 Các bộ công cụ phát triển ứng dụng Java

IDE Bộ các công cụ phần mềm để soạn thảo, biên dịch, liên kết, gỡ rối, v...v... (như VS Studio của Microsoft)

IDE giúp phát triển ứng dụng nhanh chóng và hiệu quả hơn. Đơn giản hóa quá trình phát triển phần mềm

Một số IDE dành cho lập trình Java là:

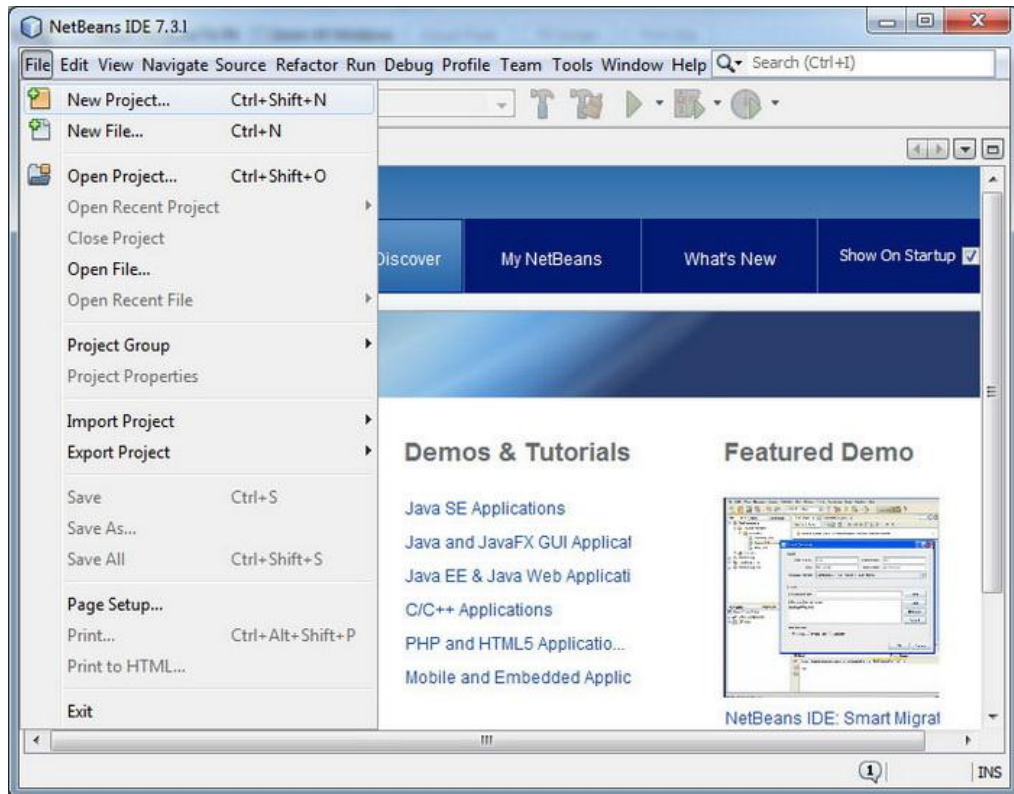
- JCreator
- NetBeans
- Eclipse
- EditPlus
- Borland Jbuilder
- Java Studio của Sun

1.5.4.2 Giới thiệu NetBeans IDE

Là môi trường phát triển – một công cụ dành cho lập trình viên để viết, 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.

NetBean IDE dễ cài đặt và chạy trên nhiều hệ điều hành: Windows, Linux, Mac OS X và Solaris.

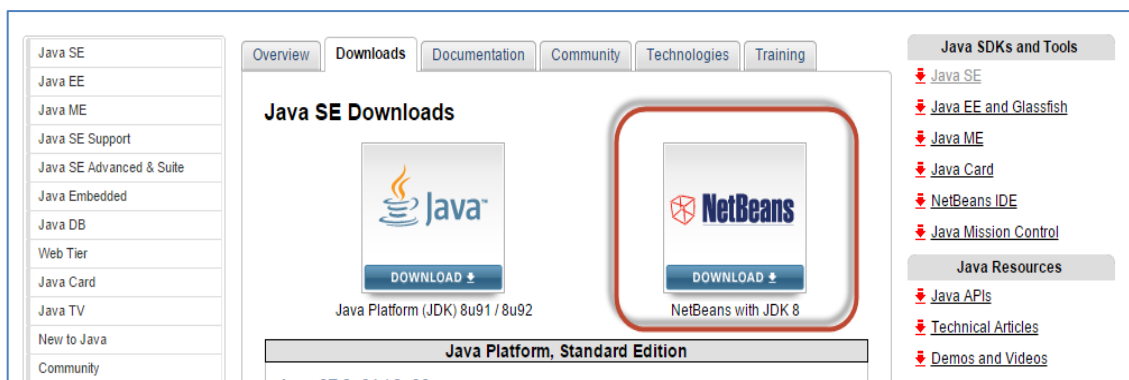


Hình 1.5: Màn hình làm việc của chương trình NvetBeans.

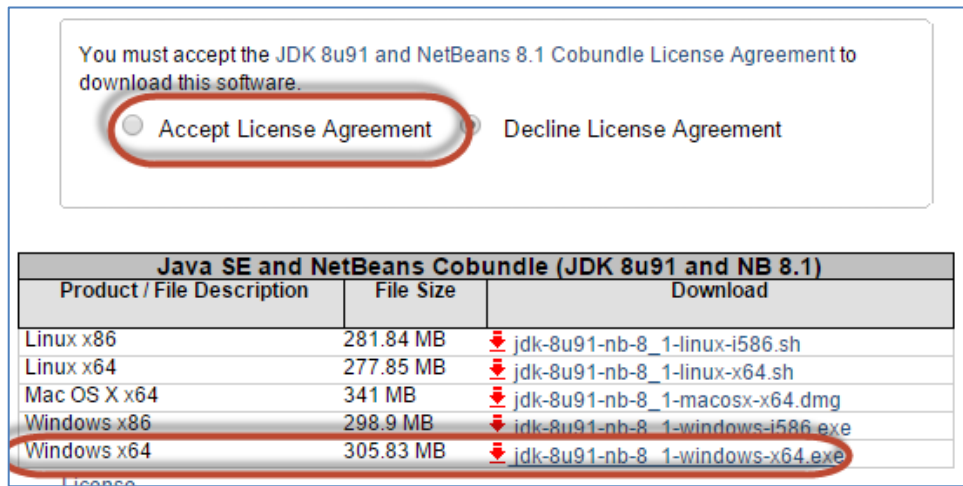
1.5.4.3 Cài đặt NetBeans IDE

Có thể cài đặt Netbeans theo 2 cách:

- Download Netbeans tích hợp JDK vài cài 1 lần
- Hoặc download JDK riêng rồi sau đó chọn lựa phiên bản Netbeans phù hợp.
- **Bước 1:** Download Netbeans tại:
 - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
 - Chọn Netbeans With JDK → chọn phiên bản phù hợp HĐH.

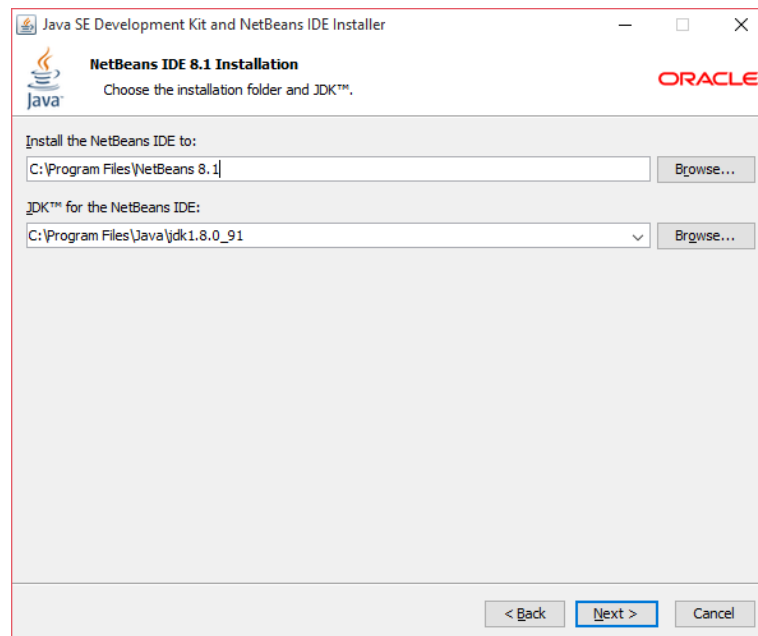


Hình 1.6: Chọn phiên bản Netbeans tích hợp JDK .



Hình 1.6: Chọn phiên bản NetBeans.

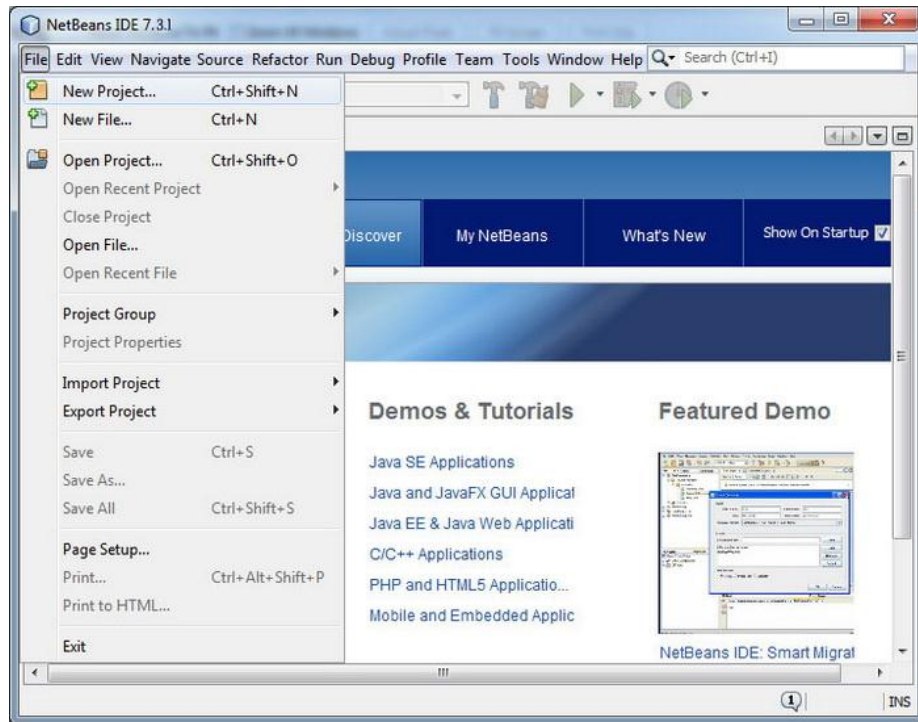
Bước 2: Tiến hành cài đặt chương trình như các chương trình thông thường khác.



Hình 1.7: Chọn thư mục cài đặt NetBeans

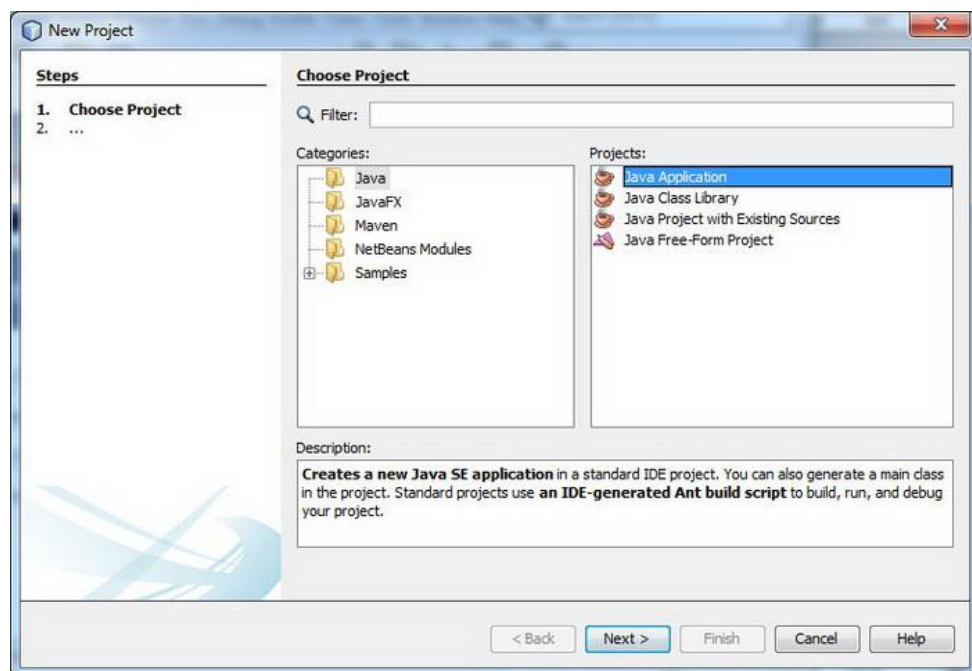
1.5.4.4 Hướng dẫn sử dụng NetBeans IDE

Mở Netbeans sau đó chọn File -> New Project



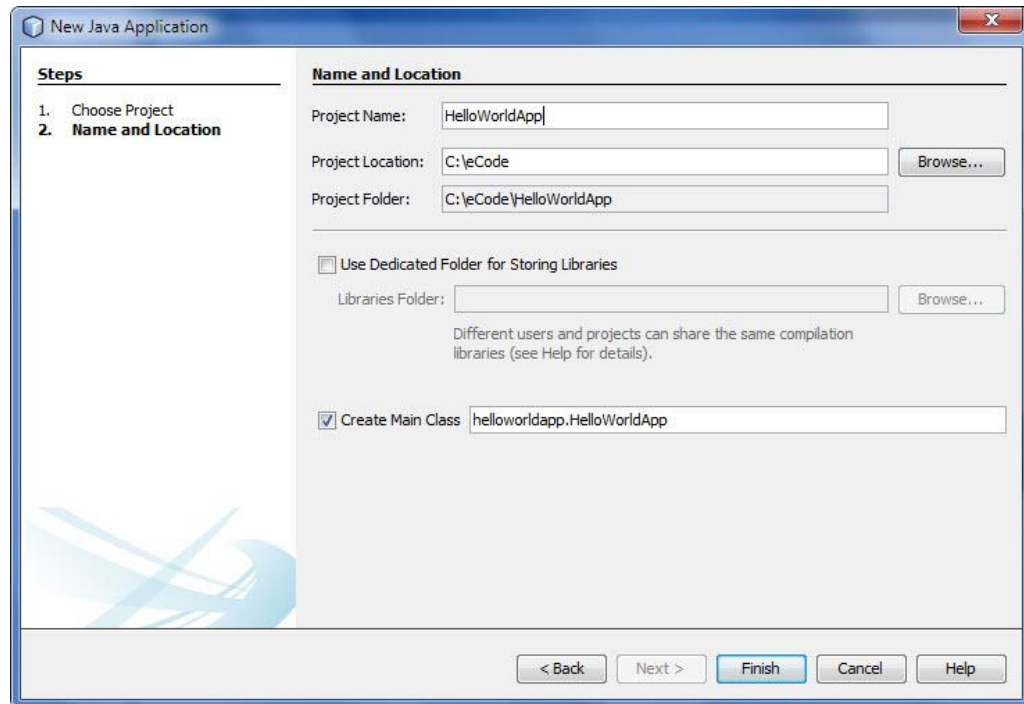
Hình 1.8: Màn hình khởi động và tạo Project mới của NetBeans

Chọn Java Application



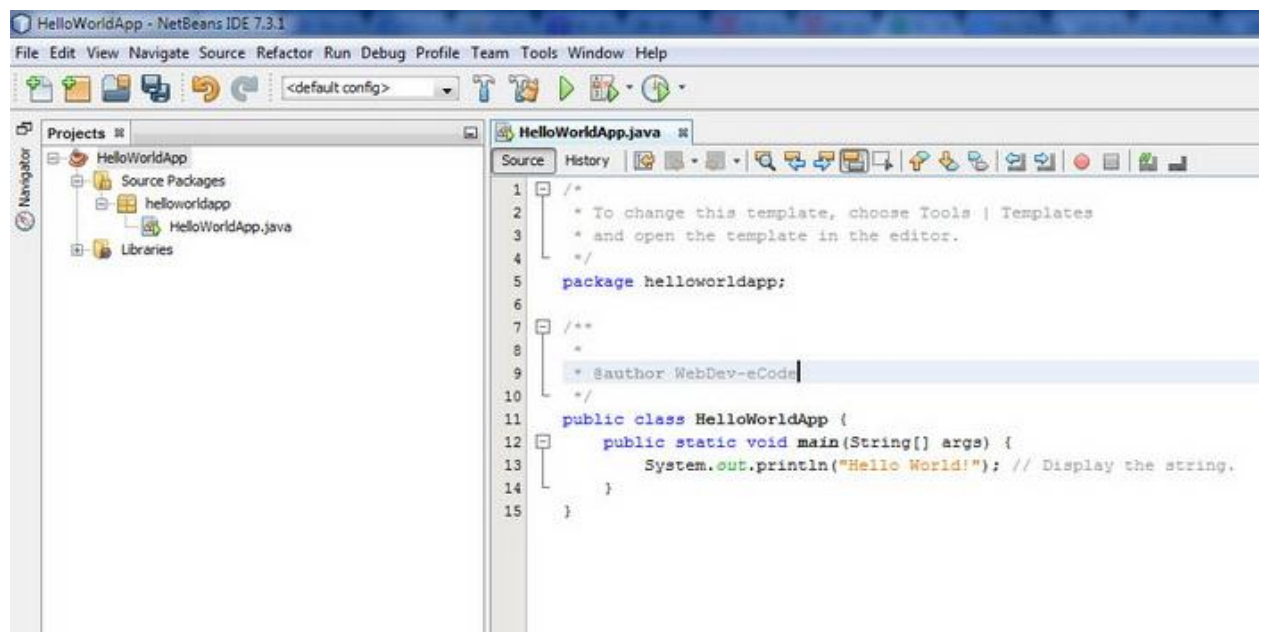
Hình 1.9: Màn hình chọn loại Project cần tạo

Điền tên Project Name và chọn đường dẫn lưu ứng dụng. Lưu ý: Tên class main trùng với tên class khai báo trong mã nguồn



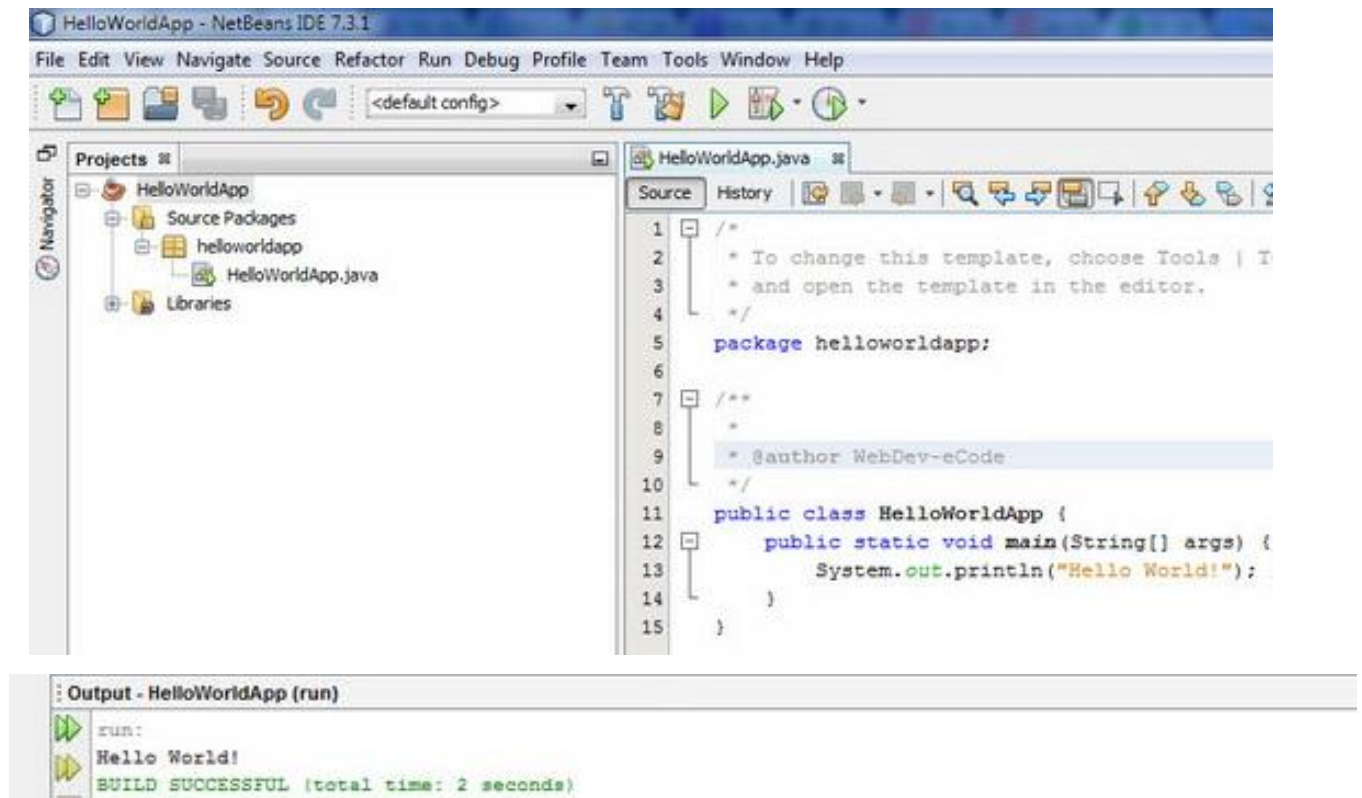
Hình 1.10: Màn hình đặt tên và chọn vị trí lưu trữ cho Project

Xóa hết mã có sẵn trong class HelloWorldApp và chèn đoạn mã sau:



Hình 1.11: Màn hình soạn thảo code cho ứng dụng

Thực thi ứng dụng ấn Shift+F6. Nếu xuất hiện thông báo màu xanh Build Success là thành công



Hình 1.12: Màn hình kết quả thực thi chương trình thành công

Một số phím thông dụng khi làm việc với Netbeans:

✓ **Tìm kiếm và thay thế**

- Ctrl-F3: tìm từ tại vị trí con trỏ chuột
- 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 projects
- Ctrl-R: đổi tên biến tại con trỏ chuột
- Alt-Shift-L: copy đường dẫn của file đang mở

✓ **Coding trong Java**

- Alt-Insert: sinh ra code
- Ctrl-Shift-I: tự động imports class cần thiết
- Alt-Shift-F: định dạng code
- Ctrl-Shift-Up/D: copy dòng tại con trỏ chuột
- Ctrl-/: bật tắt comment
- Ctrl-E/X: xóa dòng tại con trỏ chuột

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

- F9: biên dịch package/file
- F11: build project chính
- Shift-F11: xóa và build project chính

- Shift-F6: Chạy file hiện tại
- F6: Chạy project

✓ **Debugging**

- Ctrl-F5: bắt đầu debugging project chính
- Ctrl-Shift-F5: bắt đầu debugging file hiện tại
- Ctrl-Shift-F6: bắt đầu debugging file
- Ctrl-Shift-F8: đặt breakpoint mới

1.6 CẤU TRÚC CHƯƠNG TRÌNH JAVA ĐƠN GẢN

- Java được tổ chức theo lớp (class). Các lệnh và các hàm (kể cả hàm main) phải thuộc một lớp. Tên lớp chứa hàm main phải giống tên file.
- Phương thức main() sẽ được gọi đầu tiên. Mỗi chương có một phương thức main().
- Dấu mở và đóng ngoặc nhọn “{” và “}” là bắt đầu và kết thúc một khối lệnh.
- Các câu lệnh phải kết thúc bằng dấu chấm phẩy ;
- Dấu hiệu chú thích làm cho chương trình dễ hiểu hơn. Trình biên dịch sẽ bỏ qua những dòng có dấu chú thích

```

1  // Tên file : Hello.java
2  /* Tác giả : Barak Obama*/
3  public class Hello
4  {
5      //Phương thức main, điểm bắt đầu của chương trình
6      public static void main( String args[ ] )
7      {
8          System.out.println( "Hello World" );
9      } // Kết thúc phương thức main
10 } // Kết thúc lớp Hello

```

TÓM TẮT

- Giới thiệu về Ngôn ngữ lập trình Java , lịch sử ra đời và phát triển: Ngôn ngữ lập trình Java là ngôn ngữ biên dịch và thông dịch, phổ biến và mạnh mẽ phát triển nhiều loại ứng dụng, đa nền tảng.
- Các đặc trưng của Java: Đơn giản; Hướng đối tượng; Độc lập phần cứng; Mạnh; Bảo mật; Phân tán; Đa luồng; Động
- Các công nghệ của Java: J2SE, J2EE, J2ME
- Các kiểu ứng dụng của Java : Ứng dụng độc lập (console application); Ứng dụng Applets; Ứng dụng giao diện (GUI application); Ứng dụng Web (Web application); Ứng dụng nhúng; Ứng dụng cơ sở dữ liệu; Ứng dụng Game
- Môi trường phát triển ứng dụng Java – Bộ công cụ Java Development Kit
- Cài đặt và hướng dẫn sử dụng NetBeans IDE
- Cấu trúc chương trình trình Java đơn giản.

BÀI TẬP

Bài tập 1: Cài đặt Java JDK & NeBeans IDE

Bài tập 2. Viết chương trình xuất ra màn hình các thông tin sau.

“Hello! I’m <your name>.”

“This is my first java program.”

Bài tập 3. Viết chương trình: Nhập các thông tin của 1 sinh viên gồm mã sinh viên, họ tên, tuổi, năm sinh, điểm trung bình. Xuất các thông tin ra màn hình (hãy sử dụng kiểu dữ liệu thích hợp).

Hướng dẫn: Nhập dữ liệu sử dụng đối tượng Scanner trong thư viện **java.util.***

```
import java.io.*; //thư viện chứa các lớp cho việc xuất, nhập
import java.util.*; //thư viện chứa lệnh nhập sử dụng Scanner
class vd
{
    public static void main (String[] args)
    {

        Scanner x=new Scanner(System.in);
        System.out.print("Nhap ma sinh vien:");
        String masv=x.nextLine();

        //nhập họ tên: tương tự
        //...

        System.out.print("Nhap tuoi:");
        byte tuoi=x.nextByte();

        //tương tự với năm sinh, điểm trung bình
        //...

    }
}
```

Bài tập 4. Viết chương trình: Nhập vào hai số nguyên x, y. Xuất ra màn hình tổng, hiệu, tích, thương của hai số trên.

Bài tập 6. Viết chương trình: Nhập vào 2 số nguyên dương a, b. Tìm USCLN và BSCNN của a và b. Ghi chú: Viết chương trình sử dụng hàm.

Bài tập 5. Viết chương trình: Nhập số nguyên dương n, liệt kê các số nguyên tố nhỏ hơn n.

Bài tập 6. Viết chương trình: Nhập 1 mảng các số nguyên từ bàn phím. Tìm phần tử lớn nhất trong mảng.

Hướng dẫn:

- Khai báo mảng: **Kiểu_dữ_liệu Tên_biến_mảng[];**
Kiểu_dữ_liệu[] Tên_biến_mảng;
- Cấp phát bộ nhớ cho mảng: **tênBiếnMảng = new KiểuDữLiệu[sốPT];**

```
//Khai báo mảng, khai báo biến giữ số lượng phần tử cho mảng
int a[], n=0;

//nhập số lượng phần tử của mảng: nhập n
//...bạn tự code

//cấp phát số ô nhớ = số lượng phần tử sẽ dùng
a=new int[n];

//nhập giá trị cho từng phần tử trong mảng
for(int i=0; i<n; i++)
    a[i] = ...//tự code

//tìm phần tử lớn nhất trong mảng, bạn tự code :)
```

-----Hết Bài 1-----