



BỘ MÔN CÔNG NGHỆ PHẦN MỀM  
VIỆN CNTT & TT  
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



IT3100

# LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài 01. Tổng quan về OOP

# Mục tiêu bài học

- Tìm hiểu tổng quan về Công nghệ lập trình hướng đối tượng
  - Khái niệm, Ưu điểm, vai trò, ứng dụng, Sự phát triển các ngôn ngữ lập trình
- Tìm hiểu về Đối tượng và lớp
  - Đối tượng, trạng thái, hành vi
  - Lớp, thuộc tính, phương thức
  - Tương tác giữa các đối tượng (thông điệp, giao diện)
- Tìm hiểu Ngôn ngữ lập trình Java
  - Giới thiệu về Java, Nền tảng Java, Mô hình dịch Java
  - Tính năng và ứng dụng của Java
- Cài đặt môi trường lập trình

# Nội dung

1. Công nghệ hướng đối tượng (HĐT)
2. Đối tượng và lớp
3. Ngôn ngữ lập trình Java
4. Cài đặt môi trường lập trình

# 1/ Công nghệ HĐT

# 1.1. Kỹ thuật lập trình

- Kỹ thuật lập trình: Kỹ thuật thực thi một giải pháp phần mềm (cấu trúc dữ liệu + giải thuật) dựa trên nền tảng một phương pháp luận (methodology) và một hoặc nhiều ngôn ngữ lập trình phù hợp với yêu cầu đặc thù của ứng dụng
- Ngôn ngữ lập trình
  - Là ngôn ngữ được chuẩn hóa
  - Cả con người và máy tính có thể đọc và hiểu được
  - Sử dụng chương trình dịch tương ứng để biên dịch toàn bộ chương trình nguồn thành mã máy trước khi thực hiện

# 1.2. Sự phát triển của ngôn ngữ lập trình



```
RELOAD EQU 0E6H    ; defining reload constant for baudrate g
ORG 0000H           ; org directive
SJMP START         ; jump to main program

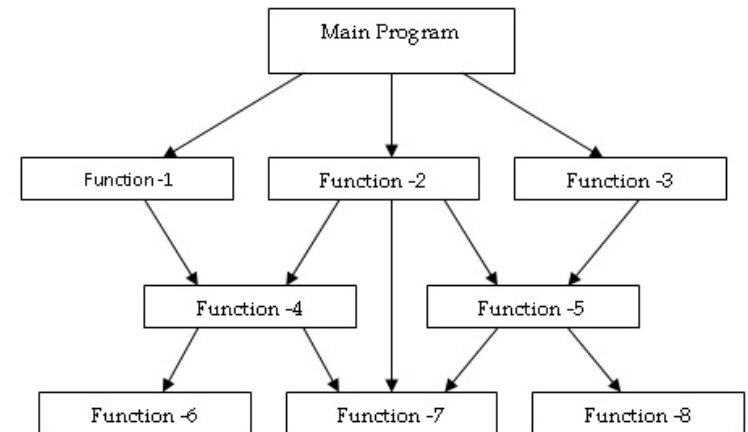
SENDCH: ORG 0023H   ; ISR for RI/TI interrupt
CLR TI             ; clear transmit flag
MOV SBUF, #'A'     ; send the ASCII value of 'A'
RETI               ; return back to main program

START: ANL PCON, #7FH ; Set SMOD=0
ANL TMOD, #0FH     ; Alter only the setting of Timer-1
ORL TMOD, #20H     ; Timer-1 in mode-2
MOV TH1, #RELOAD   ; Move the reload value to TH1
SETB TR1           ; start Timer-1 for baud rate generation
MOV SCON, #40H     ; set serial port in mode-1
ORL IE, #90H       ; Enable serial port interrupt

MOV SBUF, #'A'     ; Transmit a character

WAIT: SJMP WAIT    ; wait till interrupt occurs

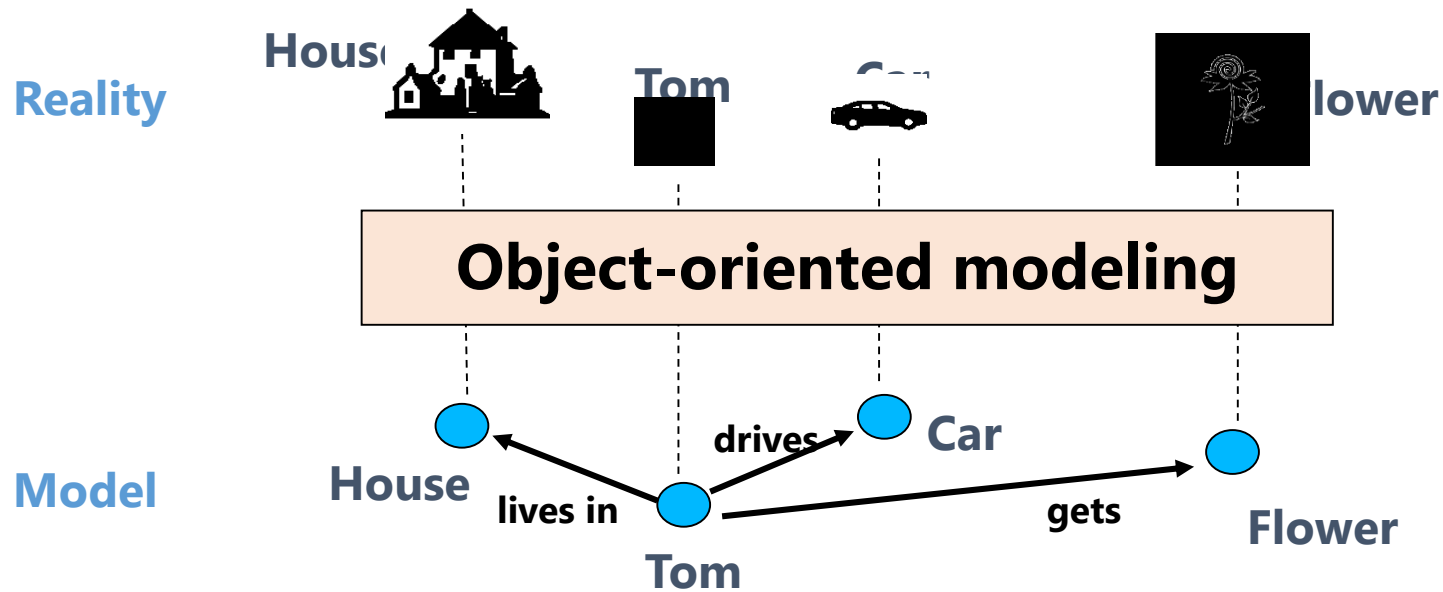
END
```



Assembly language

## 1.3. Lập trình hướng đối tượng

- Thể hiện các thành phần của bài toán là các “đối tượng” (object).
- Hướng đối tượng là một kỹ thuật để mô hình hóa hệ thống thành nhiều đối tượng tương tác với nhau

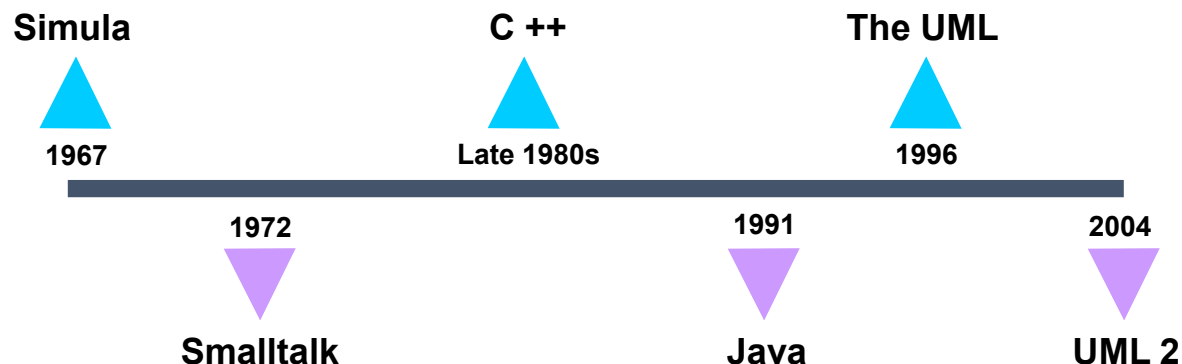


# 1.4 Công nghệ đối tượng (OOT)

- *Công nghệ đối tượng* là một tập các quy tắc (trừu tượng hóa, đóng gói, đa hình), các hướng dẫn để xây dựng phần mềm, cùng với ngôn ngữ, cơ sở dữ liệu và các công cụ khác hỗ trợ các quy tắc này.

(*Object Technology - A Manager's Guide*, Taylor, 1997)

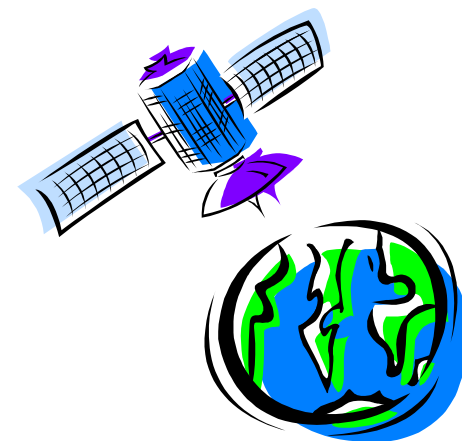
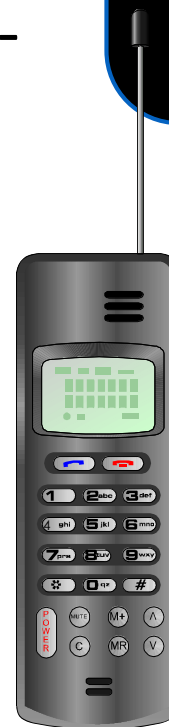
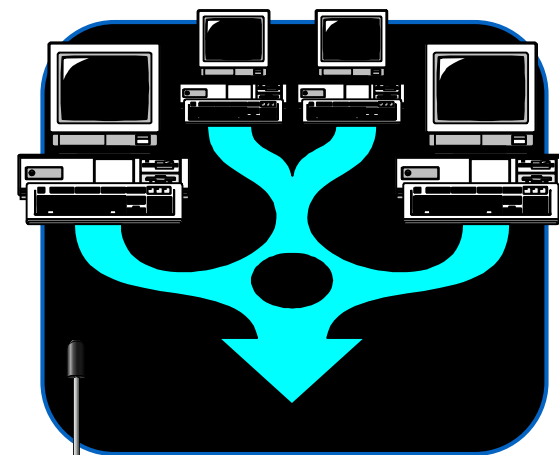
- Các mốc chính của công nghệ đối tượng



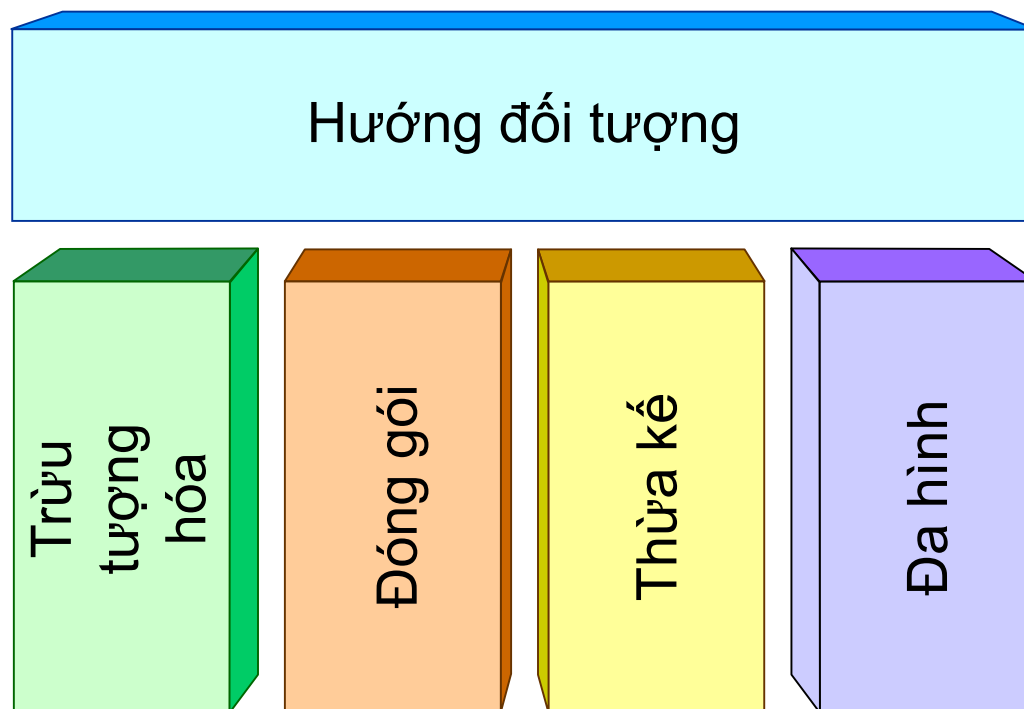


# 1.4.1 OOT được sử dụng ở đâu?

- Các hệ thống Client/Server và phát triển Web
- Hệ nhúng (embedded system)
- Hệ thống thời gian thực (real-time)
- Hệ thống phần mềm nói chung...



# 1.5. Các nguyên lý cơ bản của OOT

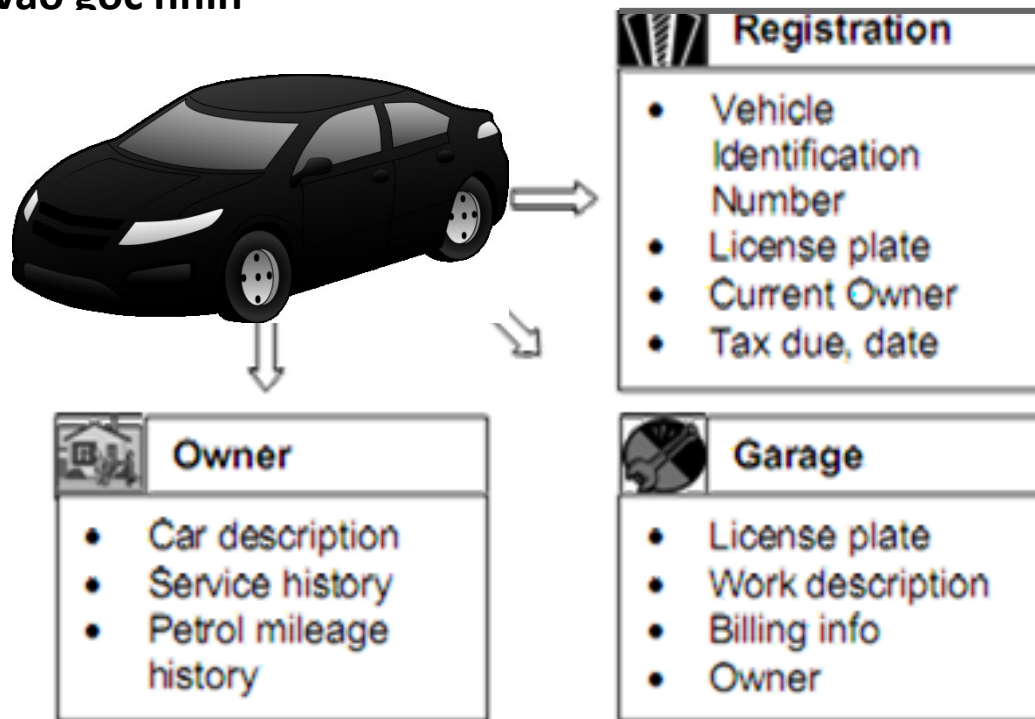


## 1.5.1 Trừu tượng hóa (Abstraction)

- Là quá trình loại bỏ đi các thông tin/tính chất cụ thể và giữ lại những thông tin/tính chất chung.
- Tập trung vào các đặc điểm cơ bản của thực thể, các đặc điểm phân biệt nó với các loại thực thể khác.
- Phụ thuộc vào góc nhìn
  - Quan trọng trong ngữ cảnh này nhưng lại không có ý nghĩa nhiều trong ngữ cảnh khác.

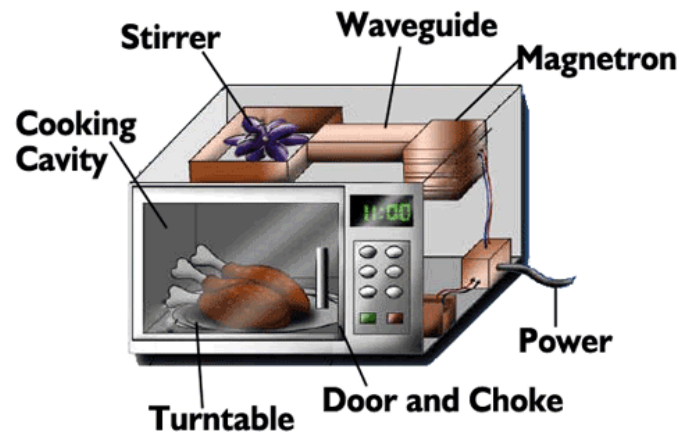
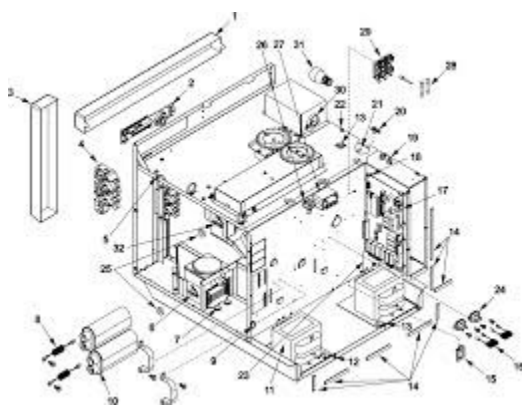
# Ví dụ: Trừu tượng hóa

Phụ thuộc vào góc nhìn



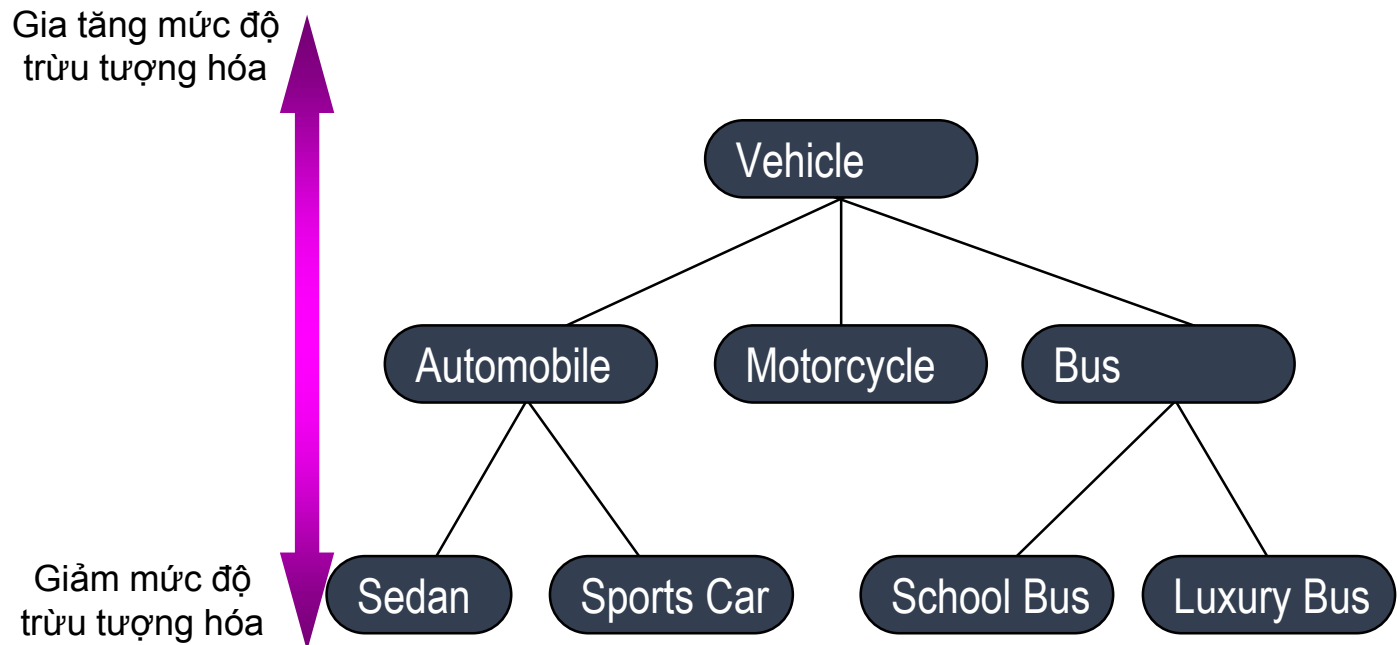
## 1.5.2 Đóng gói (Encapsulation)

- Che giấu, ẩn đi chi tiết thực hiện bên trong
- Cung cấp cho thế giới bên ngoài một giao diện
- Việc sử dụng không ảnh hưởng bởi chi tiết bên trong.



## 1.5.3. Thừa kế (Inheritance)

- Xếp hạng hay xếp thứ tự các mức trừu tượng vào một cấu trúc cây
- Các phần tử ở cùng cấp trong sơ đồ phân cấp thì có cùng mức trừu tượng hóa



## 1.5.4. Đa hình (polymorphism)

- Đa hình: “one name, many forms”
- Nạp chồng phương thức: phương thức cùng tên, nhưng hoạt động khác nhau
  - Add(int x, int y)
  - Add(float x, float y)
  - Add(float x, float y, float z)
- Ghi đè phương thức (Method Overriding)
  - Một Intern (thực tập sinh) là một Intern, đồng thời cũng có thể được xem là một Staff (nhân viên)
  - Phương thức quét thẻ của Intern khác với phương thức quét thẻ của Staff

## 2/ Đối tượng và lớp

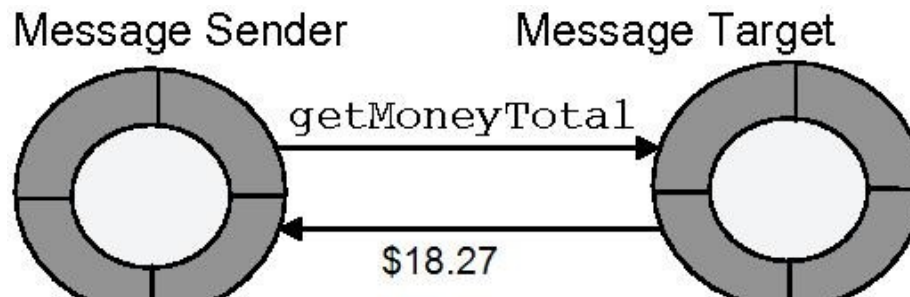


## 2.1 Đặc tính của LT HĐT

- Alan Kay đã tổng hợp các đặc tính của LT HĐT:
  1. Tất cả đều là **đối tượng**
  2. Chương trình phần mềm có thể coi là một tập hợp các đối tượng **tương tác** với nhau
  3. Mỗi đối tượng trong chương trình có các **dữ liệu độc lập** của mình và **chiếm bộ nhớ riêng** của mình.
  4. Mỗi đối tượng đều có dạng **đặc trưng** của **lớp** các đối tượng đó
  5. Tất cả các đối tượng thuộc về cùng một lớp đều có các **hành vi** giống nhau

## 2.2 Hướng cấu trúc vs. Hướng ĐT

- Hướng cấu trúc:
  - data structures + algorithms = Program
  - (cấu trúc dữ liệu + giải thuật = Chương trình)
- Hướng đối tượng:
  - objects + messages = Program
  - (đối tượng + thông điệp = Chương trình)



## 2.3 Đối tượng là gì?

- Đối tượng trong thế giới thực, là một thực thể cụ thể mà thông thường chúng ta có thể *sờ, nhìn thấy* hay *cảm nhận* được.
- Tất cả có trạng thái (state) và hành vi (behaviour)

	Trạng thái	Hành động	
<b>Con chó</b>	Tên Màu Giống Vui sướng	Sủa Vẫy tai Chạy Ăn	
<b>Xe đạp</b>	Bánh răng Bàn đạp Dây xích Bánh xe	Tăng tốc Giảm tốc Chuyển bánh răng ...	

## 2.3.1 Trạng thái và hành vi

- Trạng thái của một đối tượng là một trong các điều kiện tại đó mà đối tượng tồn tại.
- Trạng thái của một đối tượng có thể thay đổi theo thời gian.



Name: J Clark  
Employee ID: 567138  
Date Hired: July 25, 1991  
Status: Tenured  
Discipline: Finance  
Maximum Course Load: 3 classes

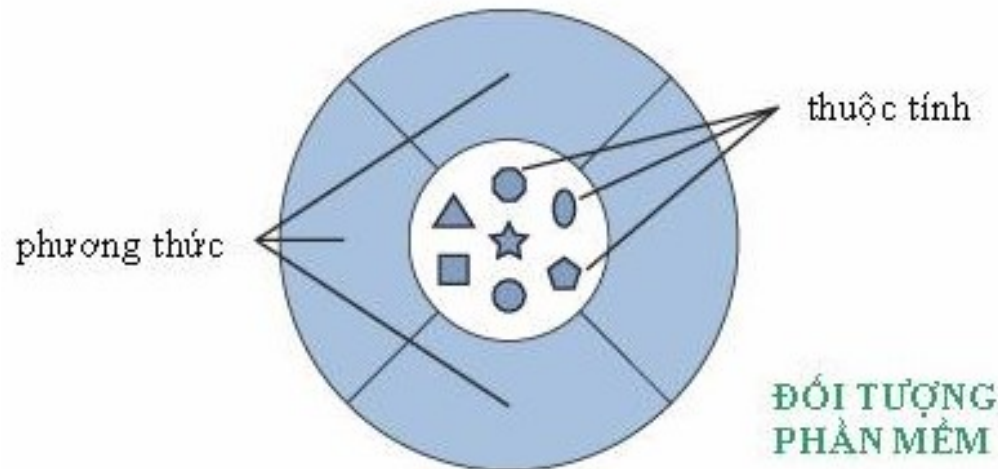
- Hành vi quyết định đối tượng đó hành động và đáp trả như thế nào đối với bên ngoài.
- Hành vi nhìn thấy được của một đối tượng được mô hình thành một tập các thông điệp nó có thể đáp trả (các thao tác mà đối tượng đó thực hiện).



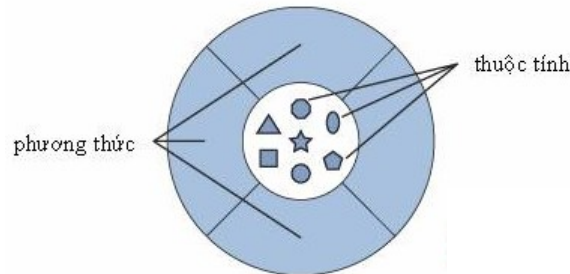
Professor Clark's behavior  
Submit Final Grades  
Accept Course Offering  
Take Sabbatical

## 2.4 Đối tượng phần mềm

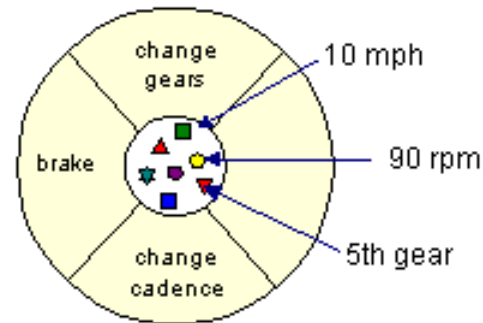
- Các **đối tượng phần mềm** được dùng để *biểu diễn* các đối tượng thế giới thực.
- Cũng có trạng thái và hành vi
  - Trạng thái: **thuộc tính** (attribute; property)
  - Hành vi: **phương thức** (method)



# Ví dụ đối tượng phần mềm



Đối tượng phần mềm



Đối tượng phần mềm **Xe Đạp**

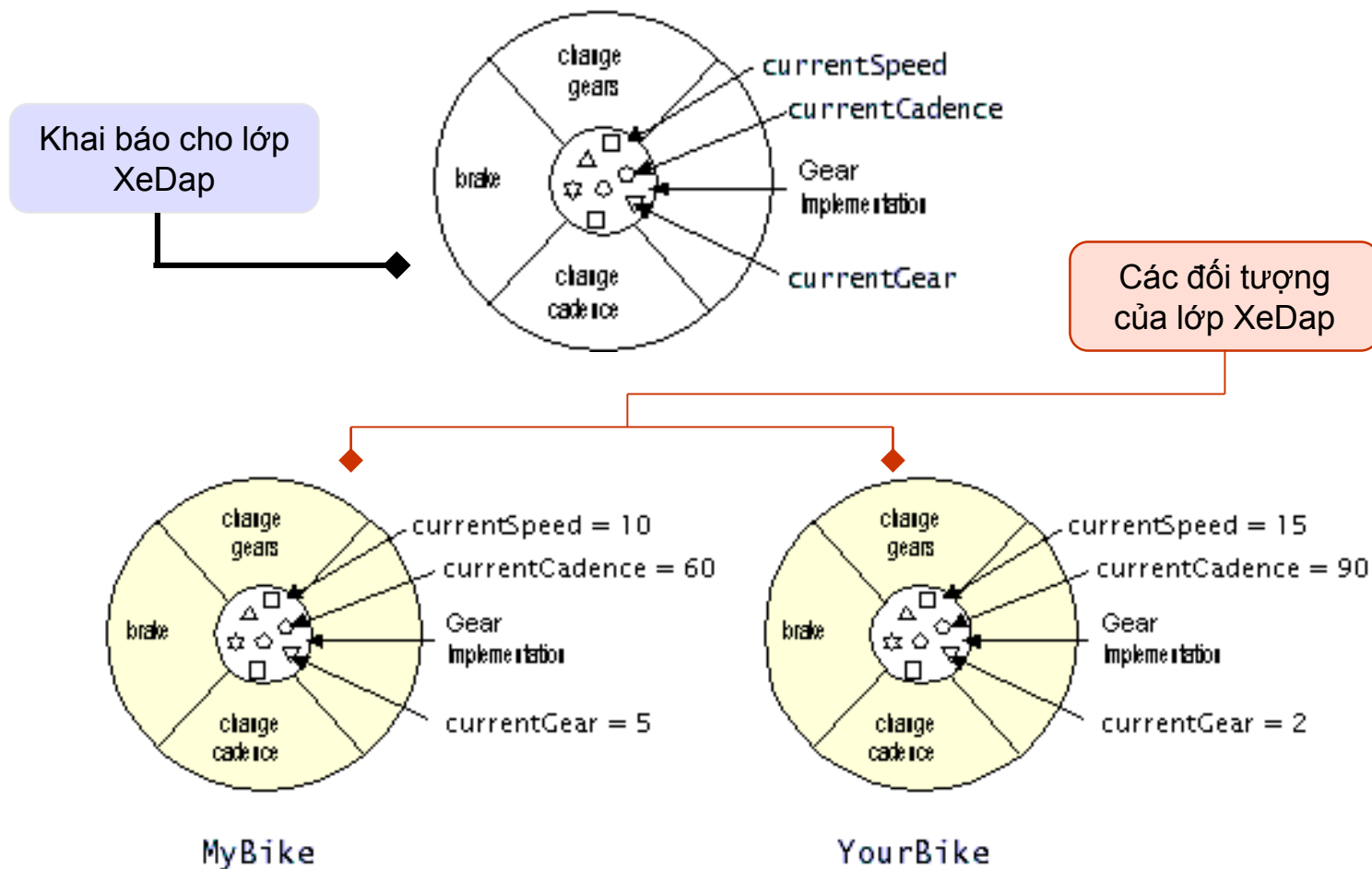
Đối tượng (object) là một thực thể phần mềm bao bọc các **thuộc tính** và các **phương thức** liên quan.

Thuộc tính được xác định bởi giá trị cụ thể gọi là **thuộc tính thể hiện**.  
Một đối tượng cụ thể được gọi là một **thể hiện**.

## 2.5 Lớp đối tượng

- Một **lớp** là một thiết kế (blueprint) hay mẫu (prototype) cho các đối tượng cùng kiểu
  - Ví dụ: lớp XeDap là một thiết kế chung cho nhiều đối tượng xe đạp được tạo ra
- Lớp định nghĩa các thuộc tính và các phương thức chung cho tất cả các đối tượng của cùng một loại nào đó
- Một **đối tượng** là một thể hiện cụ thể của một lớp.
  - Ví dụ: mỗi đối tượng xe đạp là một thể hiện của lớp XeDap
- Mỗi thể hiện có thể có những thuộc tính thể hiện khác nhau
  - Ví dụ: một xe đạp có thể đang ở bánh răng thứ 5 trong khi một xe khác có thể là đang ở bánh răng thứ 3.

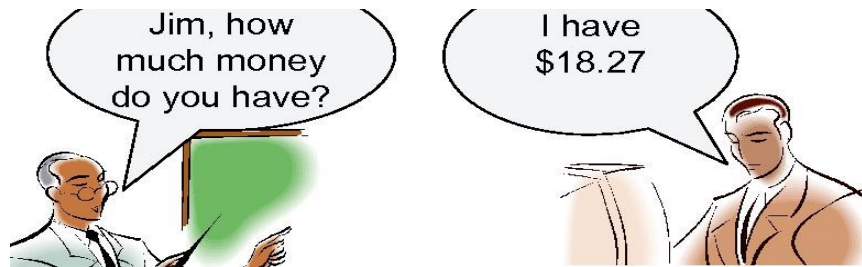
# Ví dụ Lớp Xe đạp



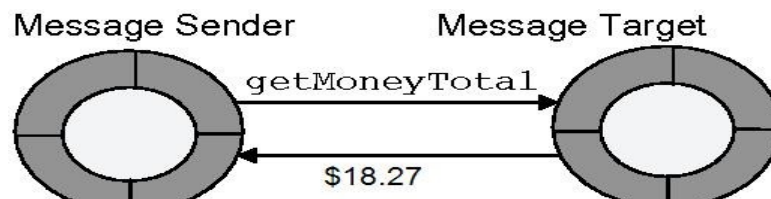


## 2.6 Tương tác giữa các đối tượng

- Sự giao tiếp giữa các đối tượng trong thế giới thực:



- Các đối tượng và sự tương tác giữa chúng trong lập trình
  - Các đối tượng giao tiếp với nhau bằng cách gửi thông điệp (message)



# Gọi hàm vs. Gửi thông điệp

- Gọi hàm (Call function)
  - Chỉ ra chính xác đoạn mã nào sẽ được thực hiện.
  - Chỉ có duy nhất một sự thực thi của một hàm với một tên nào đó.
  - Không có hai hàm trùng tên.
- Gửi thông điệp
  - Yêu cầu một dịch vụ từ một đối tượng và đối tượng sẽ quyết định cần phải làm gì.
  - Các đối tượng khác nhau sẽ có các cách thực thi các thông điệp theo cách khác nhau.

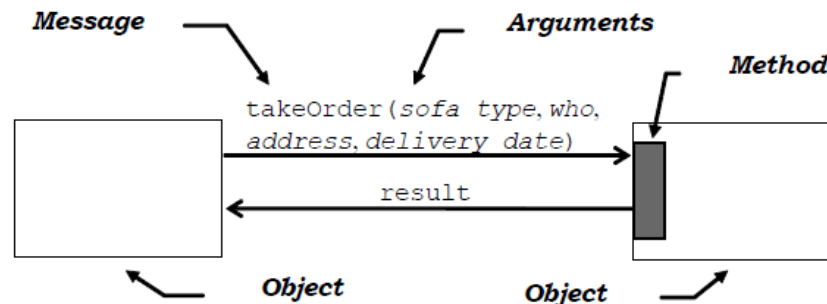
# Thông điệp vs. Phương thức

- Thông điệp

- Được gửi từ đối tượng này đến đối tượng kia, không bao gồm đoạn mã thực sự sẽ được thực thi.

- Phương thức

- Thủ tục/hàm trong ngôn ngữ lập trình cấu trúc.
- Là sự thực thi dịch vụ được yêu cầu bởi thông điệp.
- Là đoạn mã sẽ được thực thi để đáp ứng thông điệp được gửi đến cho đối tượng.



3/ NNLT Java

## 3.1. Java là gì?



- Java là một ngôn ngữ lập trình HĐT được phát triển bởi Sun Microsystems, nay thuộc sở hữu của Oracle.
- Ban đầu được sử dụng để xây dựng ứng dụng điều khiển các bộ xử lý bên trong các thiết bị điện tử dân dụng như máy điện thoại cầm tay, lò vi sóng...
- Java = 1 ngôn ngữ lập trình = 1 công nghệ = 1 nền tảng phát triển

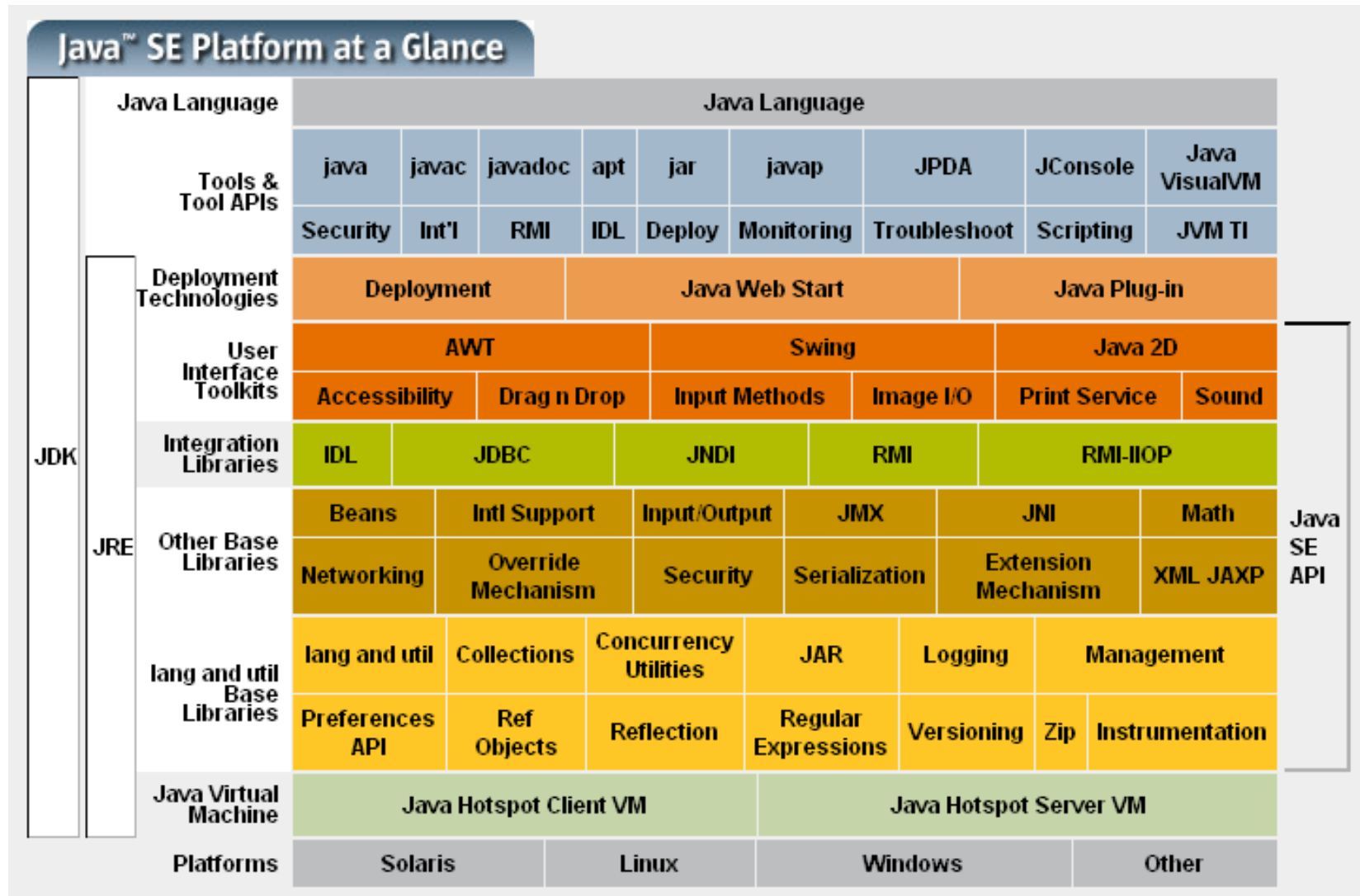


Green Team and James Gosling  
(the leader)

## 3.2 Nền tảng Java

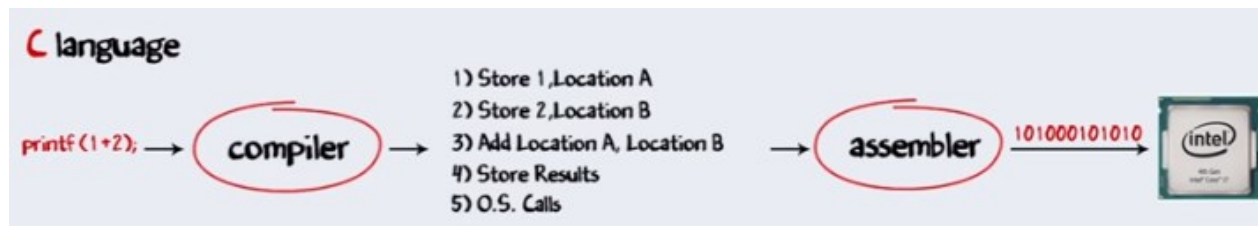
- Mỗi sản phẩm Java gồm:
  - Bộ công cụ phát triển: J2SDK, JDK hay SDK (development kit)
    - JDK = JRE + tools tiện ích (\*.exe) + tài liệu + thư viện
  - Môi trường chạy JRE (runtime environment): môi trường thực thi
    - JRE = JVM + Java runtime library (trên desktop)
    - Server JRE: JRE trên server
  - Máy ảo java JVM (virtual machine)
  - Các đặc tả chi tiết kỹ thuật, Ngôn ngữ lập trình, Các công nghệ đi kèm
- Java hỗ trợ các platform
  - Sun Solaris, Linux, Mac OS, FreeBSD & Microsoft Windows.

# Ví dụ các thành phần trong Java SE



## 3.3 Mô hình dịch của Java

- Mô hình biên dịch truyền thống
  - Mã nguồn được biên dịch thành mã nhị phân



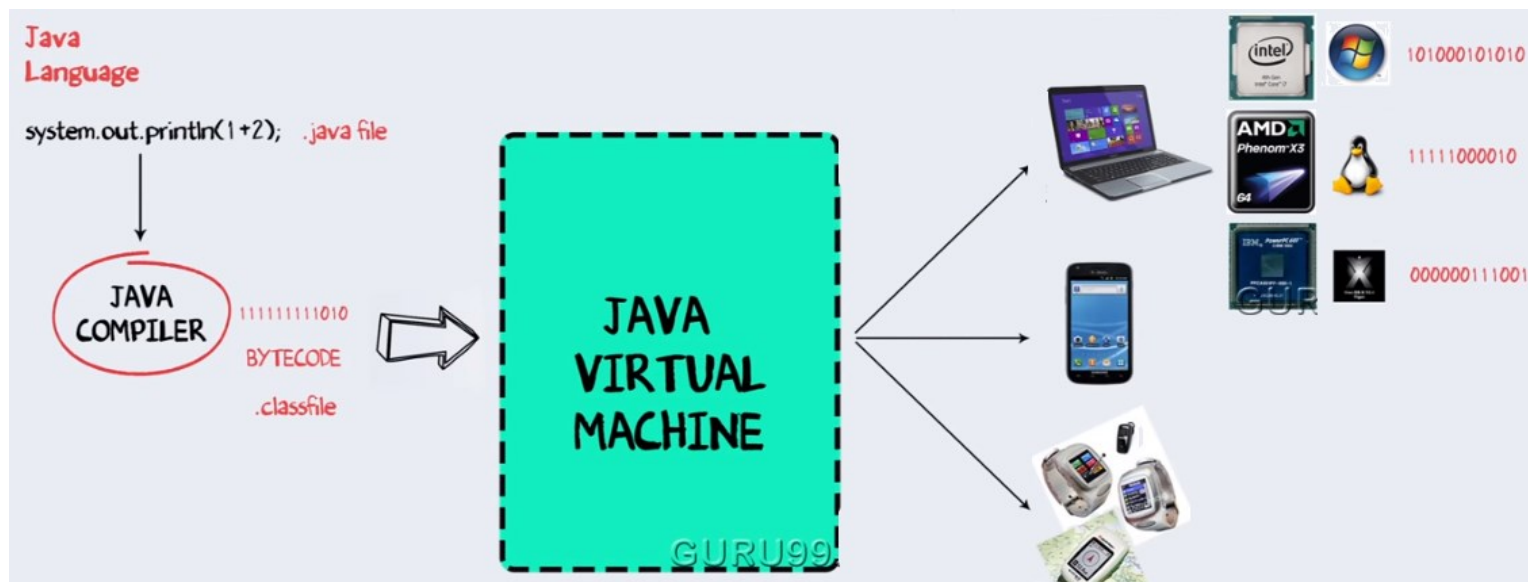
- trên các platform khác nhau là khác nhau





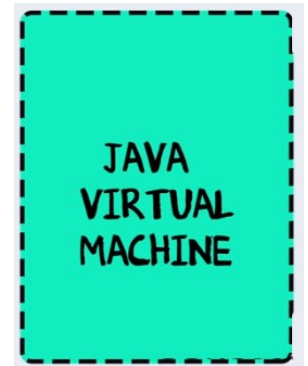
## 3.3 Mô hình dịch của Java (2)

- Mô hình dịch của Java
  - Tiêu chí "Viết (code) một lần, thực thi khắp nơi" ("Write Once, Run Anywhere" (WORA))
  - Mã nguồn được biên dịch thành bytecode rồi được thông dịch bởi JVM



## 3.3 Mô hình dịch của Java (3)

- Máy ảo Java (Java Virtual Machine):
  - Máy ảo Java là trái tim của ngôn ngữ Java
  - Tạo ra môi trường bên trong để thực thi lệnh:
    - Nạp các file .class
    - Quản lý bộ nhớ
    - Dọn “rác”
  - Trình thông dịch “**Just In Time** - JIT”
    - Chuyển tập lệnh bytecode thành mã máy cụ thể cho từng loại CPU.



## 3.4. Tính năng của Java

- Đơn giản, hướng đối tượng và quen thuộc.
- Mạnh mẽ và an toàn.
- Kiến trúc trung lập và di động.
- Thực thi với hiệu suất cao.
- Thông dịch, đa luồng và động.
- Dễ sử dụng cho người dùng Java.

## 3.4. Tính năng của Java (2)

- Đơn giản
  - Từ khóa
    - Java có 50 từ khóa
      - So với Cobol hay VB có tới hàng trăm từ khóa
    - Có ý nghĩa đặc biệt trong ngôn ngữ
    - Được sử dụng để viết các câu lệnh
- Hướng đối tượng
  - Java hỗ trợ phát triển phần mềm bằng cách sử dụng khái niệm “đối tượng”
  - Phần mềm được phát triển sử dụng Java bao gồm các lớp và các đối tượng

## 3.4. Tính năng của Java (3)

- Mạnh mẽ
  - Thư viện lớp: Hàng trăm lớp được viết trước với nhiều các phương thức tiện ích.
  - Java sử dụng mô hình con trỏ không cho phép truy cập trực tiếp vào bộ nhớ; bộ nhớ không thể ghi đè.
- An toàn
  - Java authentication dựa vào các phương pháp mã hóa khóa công khai
  - Mô hình con trỏ Java bảo vệ dữ liệu riêng tư trong các đối tượng và ngăn các ứng dụng chưa được authorized không được phép truy cập cấu trúc dữ liệu

## 3.4. Tính năng của Java (4)

- Kiến trúc trung lập, di động
  - Hỗ trợ nhiều platform, "Write Once, Run Anywhere"
- Network capable
  - Hỗ trợ phát triển các ứng dụng trong môi trường mạng
  - Java hỗ trợ phát triển các ứng dụng phân tán
  - Lý tưởng cho các ứng dụng Web
- Thực thi với hiệu suất cao

## 3.4. Tính năng của Java (5)

- Thông dịch
  - Chương trình nguồn \*.java được biên dịch thành \*.class và sau đó sẽ được thông dịch thành mã máy
- Đa luồng (Multi-threaded)
  - Cho phép chương trình của bạn chạy nhiều hơn một tác vụ tại cùng một thời điểm.
- Khả chuyển (Portable)
  - Các chương trình có thể viết và biên dịch một lần, rồi chạy trên các nền tảng khác
  - Nhờ mô hình biên dịch/thông dịch (WORE – Write Once, Run Everywhere)

## 3.5. Các ứng dụng của Java

Java Card	ƯD trên thẻ thông minh, thẻ SIM
Oracle Java Cloud Service	ƯD trên Cloud
Java Platform, Enterprise Edition	ƯD quy mô doanh nghiệp
Oracle Java Embedded	ƯD trên thiết bị nhúng
Oracle Java ME Embedded Client	ƯD trên thiết bị nhúng giới hạn tài nguyên
Oracle Java ME Software Development Kit	ƯD trên Windows sử dụng thiết bị nhúng
Java SE	Phiên bản chuẩn, dành cho mọi ứng dụng
Oracle Java SE Embedded	Phiên bản chuẩn, dành cho ứng dụng nhúng
Java TV	ƯD trên TV thông minh



## 4/ Cài đặt môi trường LT

# 4.1 Các bước cài đặt Java

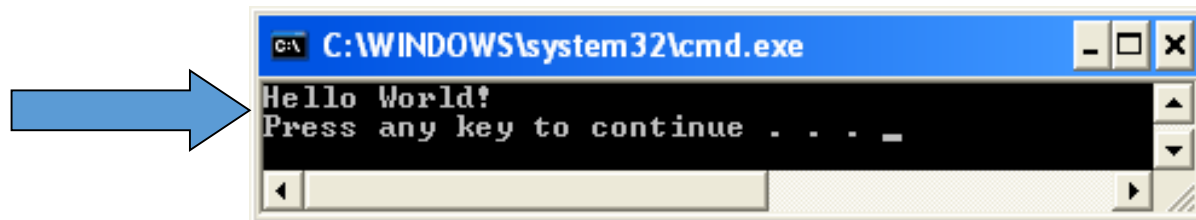
- Bước 1: Cài đặt JDK
- Bước 2: Cài đặt các biến môi trường (nếu dùng cmd)
- Bước 3: Cài trình soạn thảo hoặc IDE (Integrated Development Environment )
  - TextPad/JCreator/NetBean/Eclipse...
- Bước 4: Lập trình/Viết mã nguồn (HelloWorld.java)
- Bước 5: Dịch
  - Gõ lệnh: `javac HelloWorld.java`
- Bước 6: Chạy chương trình
  - Gõ lệnh: `java HelloWorld.class`

## 4.1 Các bước cài đặt Java (2)

- Trong Windows
  - JAVA\_HOME = C:\Program Files\Java\jdk1.6
  - PATH = ...;% JAVA\_HOME%\bin;
  - CLASSPATH = C:\Program Files\Java\jdk1.6\lib;.;  
C:\Program Files\Java\jdk1.6\include
- Trong Linux
  - JAVA\_HOME=/usr/lib/jvm/java-7-sun
  - PATH=\$PATH:\$JAVA\_HOME/bin

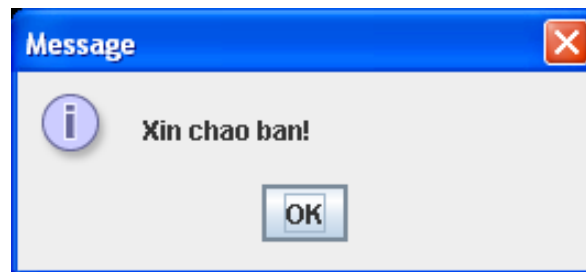
## 4.2. Chương trình ví dụ

```
// HelloWorld.java  
// Chương trình hiển thị dòng chữ "Hello World"  
public class HelloWorld {  
    public static void main(String args[]){  
        System.out.println( "Hello World!" );  
    }  
}
```



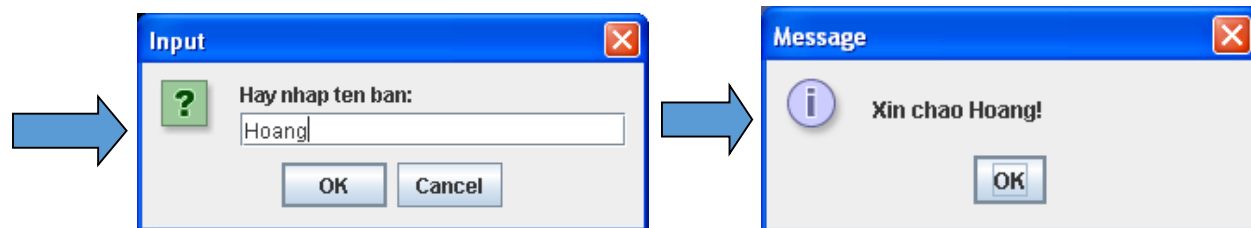
## 4.2. Chương trình ví dụ (2)

```
import javax.swing.JOptionPane;  
public class FirstDialog{  
    public static void main(String[] args){  
        JOptionPane.showMessageDialog(null,  
                                     "Xin chao ban!");  
        System.exit(0);  
    }  
}
```



## 4.2. Chương trình ví dụ (3)

```
import javax.swing.JOptionPane;  
public class HelloNameDialog{  
    public static void main(String[] args){  
        String result;  
        result = JOptionPane.showInputDialog("Hay nhap  
                                                ten ban:");  
        JOptionPane.showMessageDialog(null,  
                                      "Xin chao "+ result + " !");  
        System.exit(0);  
    }  
}
```



# Bài tập

**Bài 1.** Cài đặt Java.

**Bài 2.** Cài đặt một trình IDE nào đó .

**Bài 3.** Chạy lại 3 ví dụ trong phần 4.