



**School of Communication & Information Technology**  
**Hanoi University of Science & Technology**

---

# **Object-Oriented Language & Theory**

## **Lec01 – Introduction to Object Orientation**

Nguyễn Thị Thu Trang  
trangntt@soict.hust.edu.vn

# Nội dung

---

1. Lập trình hướng đối tượng
2. Công nghệ hướng đối tượng
3. Ngôn ngữ mô hình hóa thống nhất UML
4. Ngôn ngữ lập trình Java
5. Ví dụ và bài tập

# Nội dung

---



1. Lập trình hướng đối tượng

2. Công nghệ hướng đối tượng

3. Ngôn ngữ mô hình hóa thống nhất UML

4. Ngôn ngữ lập trình Java

5. Ví dụ và bài tập

# 1.1. Lịch sử phát triển của các NNLT

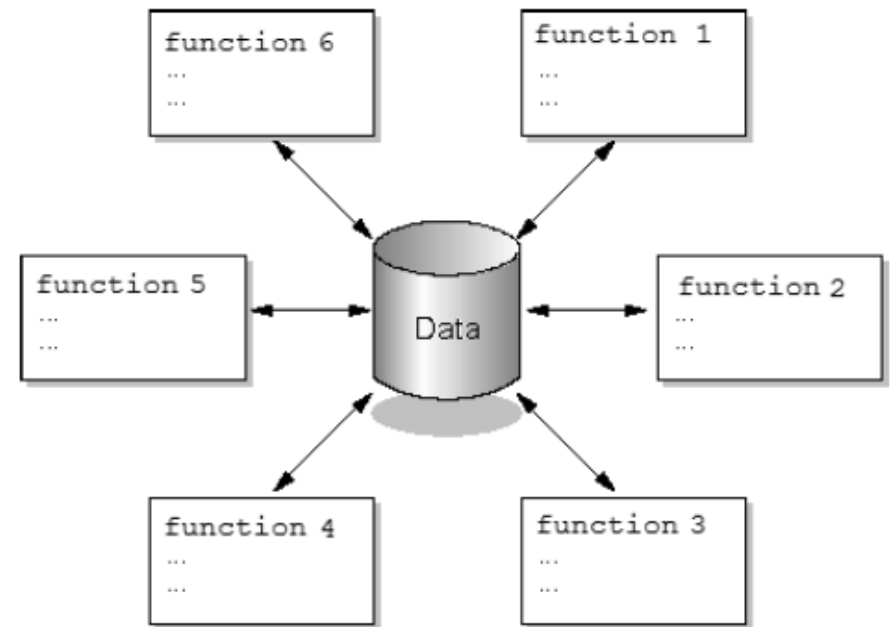
---

- a. Hợp ngữ (Assembly language):
  - Là một ngôn ngữ lập trình tuần tự, gần với tập các lệnh mã máy của CPU.
  - Khó nhớ, khó viết, nhất là với những bài toán phức tạp.
  - Khó sửa lỗi, bảo trì.

# 1.1. Lịch sử phát triển của các>NNLT (2)

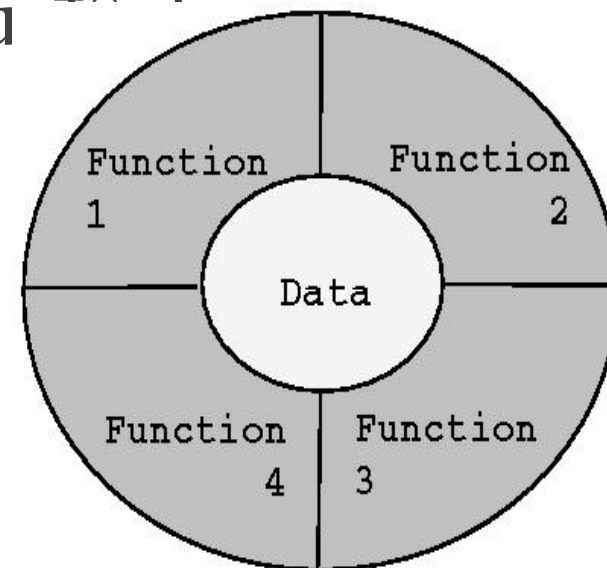
- b.>NNLT cấu trúc/thủ tục:

- Xây dựng chương trình dựa trên các hàm/thủ tục/chương trình con
- Dữ liệu và xử lý (hàm) tách rời nhau
- Các hàm không bắt buộc phải tuân theo một cách thức chung truy cập vào dữ liệu



## 1.1. Lịch sử phát triển của các>NNLT (3)

- c.>NNLT 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



## 1.2. Đối tượng là gì?

---


- Đối tượng trong thế giới thực
  - Ví dụ một chiếc ô tô
- Liên quan đến chiếc ô tô:
  - Các thông tin về chiếc xe như: màu sắc, tốc độ, số km đã đi được,...
  - Các hoạt động của chiếc xe như: tăng tốc khi nhấn ga, giảm tốc khi đạp phanh,...



# Đối tượng thế giới thực

- Một **đối tượng thế giới thực** là một thực thể cụ thể mà thông thường chúng ta có thể *sờ, nhìn thấy* hay *cảm nhận* được.

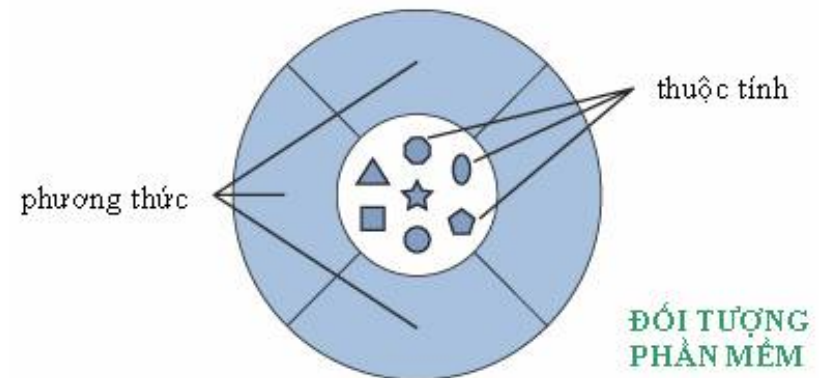
- Tất cả có trạng thái (state) và hành động (behaviour)

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



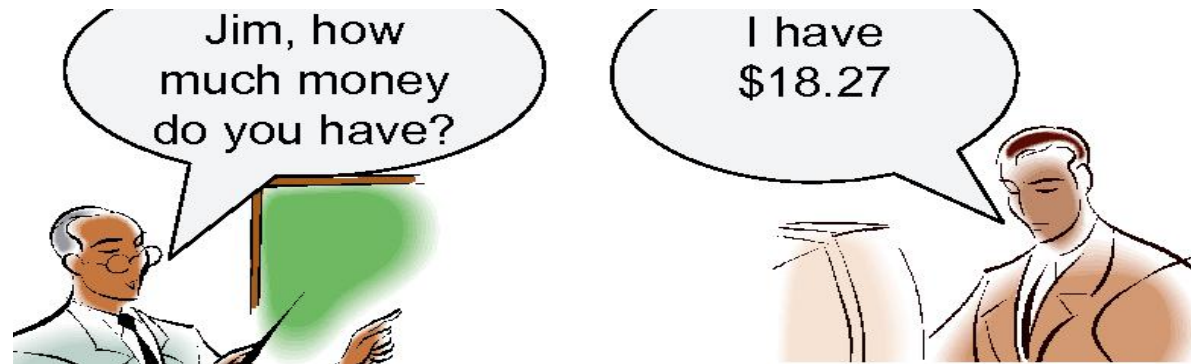
# Đối tượng phần mềm

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

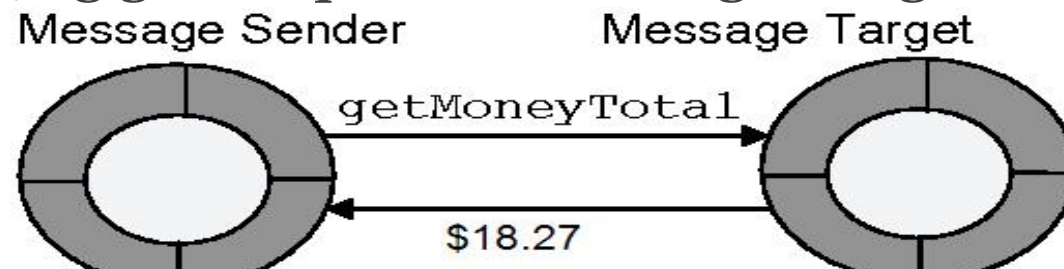


# Tương tác giữa các đối tượng

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

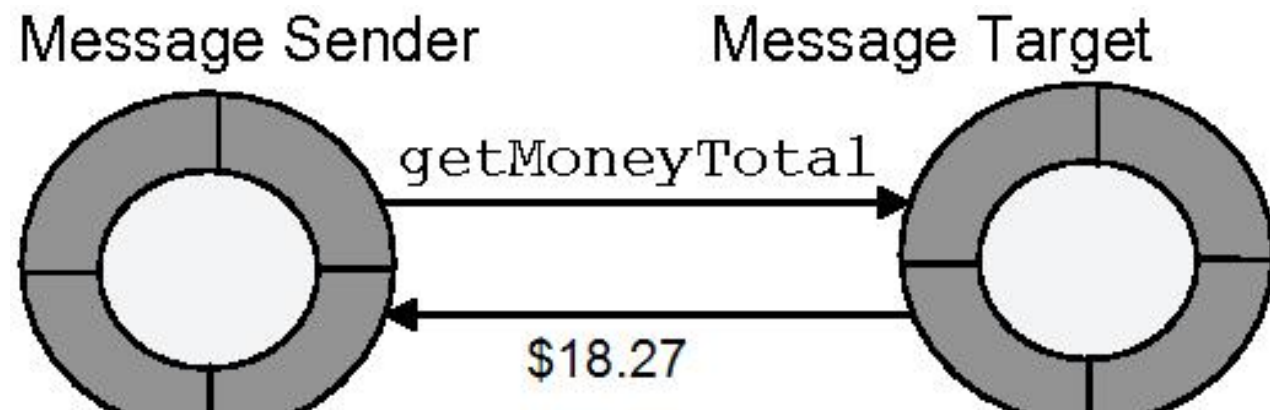


- Các đối tượng và sự tương tác giữa chúng trong lập trình
  - Các đối tượng giao tiếp với nhau bằng cách gửi thông điệp (message)



# 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)



# Nội dung

---

1. Lập trình hướng đối tượng

→ 2. Công nghệ hướng đối tượng

3. Ngôn ngữ mô hình hóa thống nhất UML

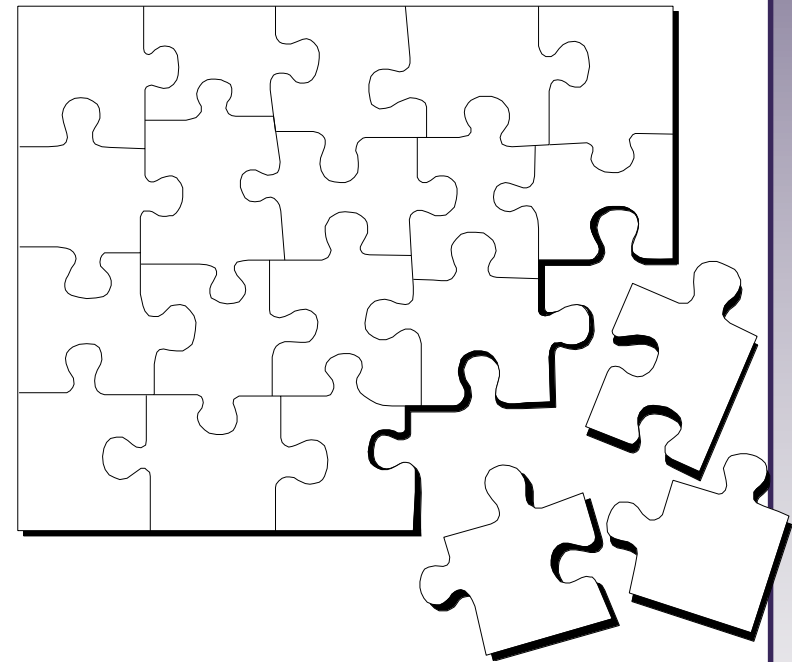
4. Ngôn ngữ lập trình Java

5. Ví dụ và bài tập

## 2. Công nghệ đối tượng (OT)

---

- 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.

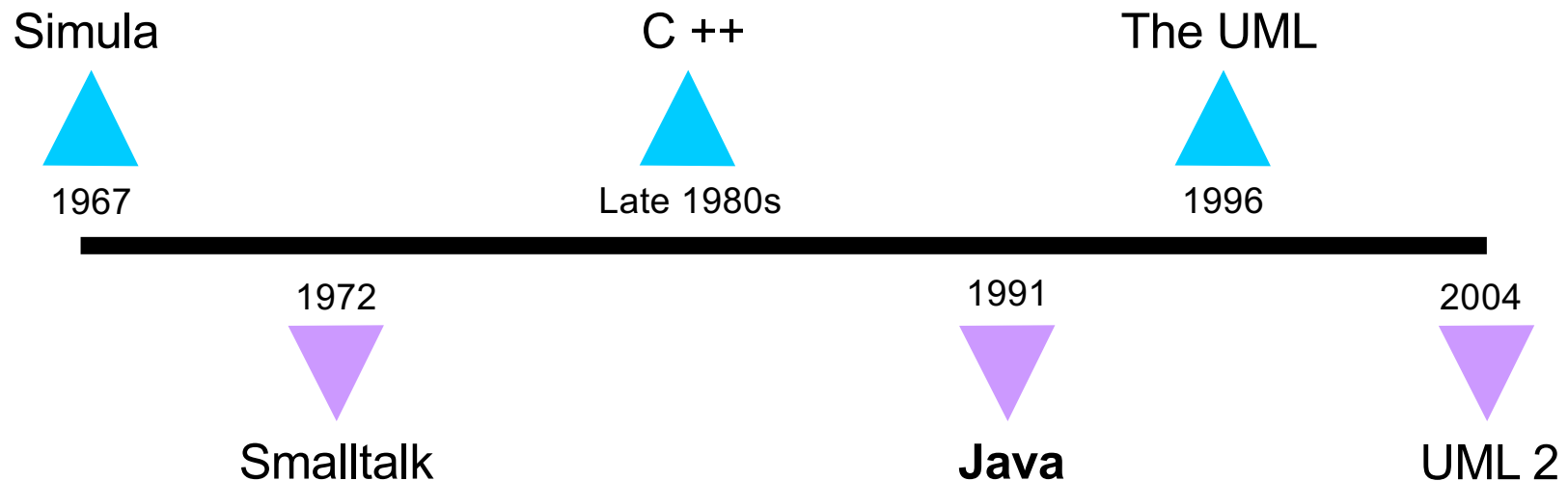


*(Object Technology - A Manager's Guide, Taylor, 1997)*

## 2.1. Lịch sử phát triển của OT

---

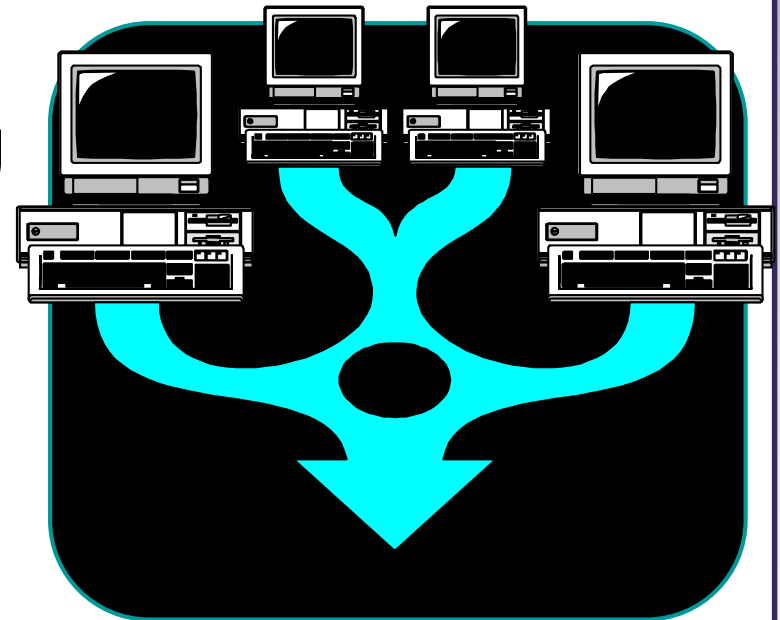
- Các mốc chính của công nghệ đối tượng



## 2.2. OT được sử dụng ở đâu?

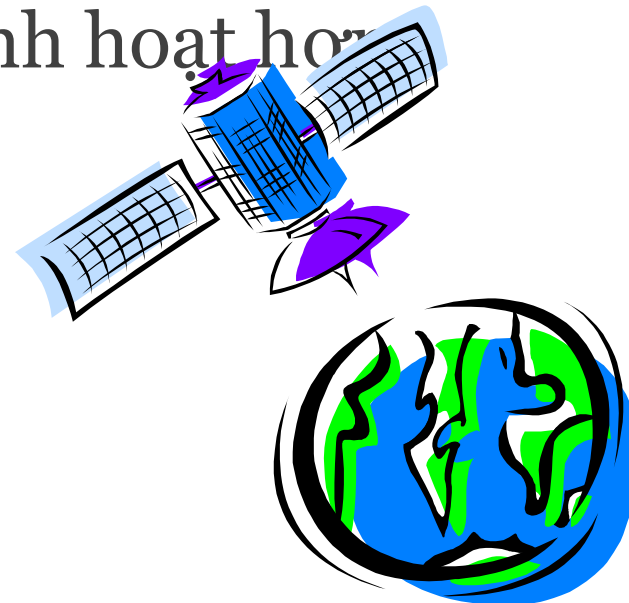
---

- Các hệ thống Client/Server và phát triển Web
  - Công nghệ đối tượng cho phép các công ty đóng gói thông tin doanh nghiệp trong các đối tượng và giúp phân phối quá trình xử lý qua mạng Internet hoặc một mạng máy tính.



## 2.2. OT được sử dụng ở đâu? (2)

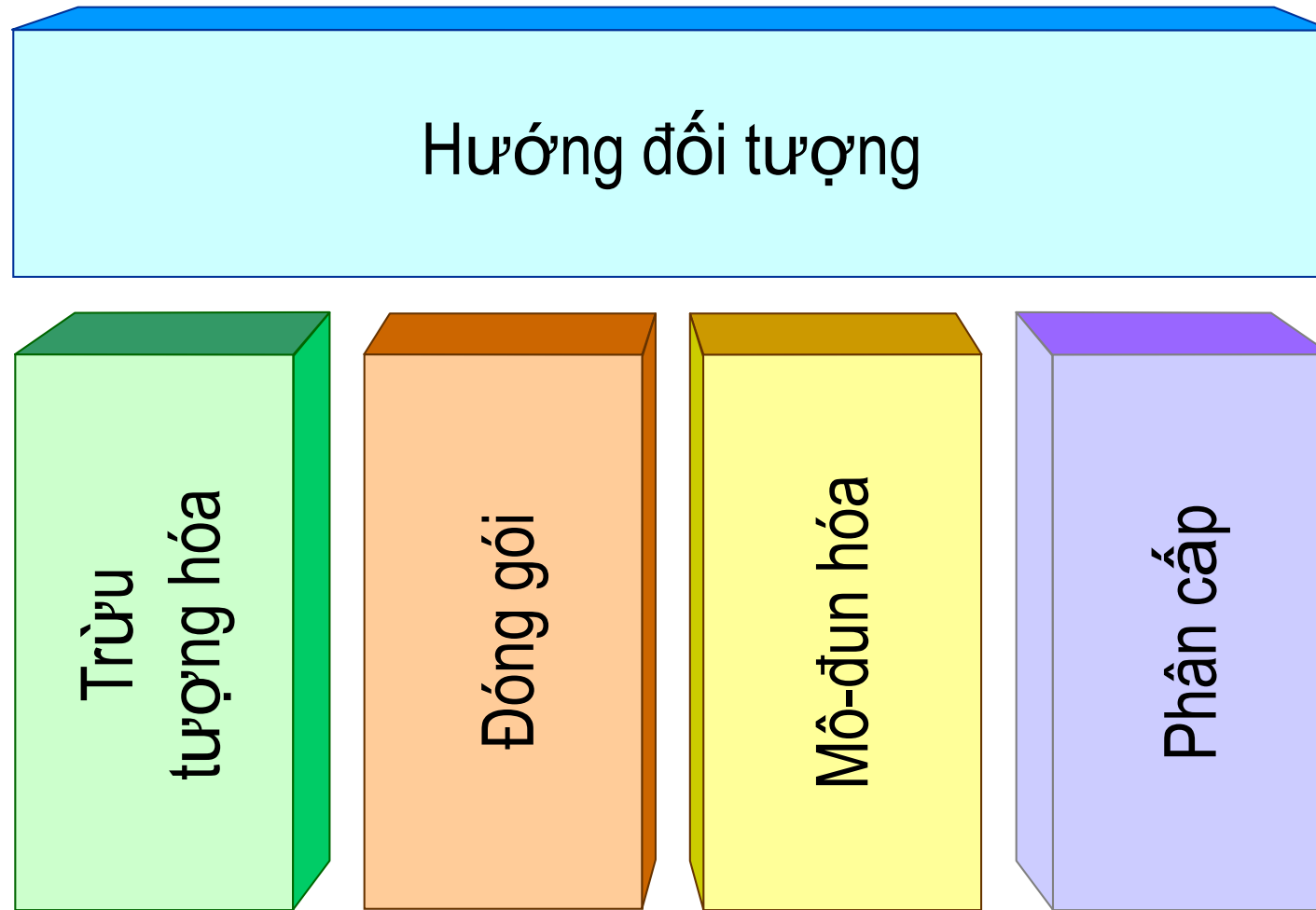
- Hệ nhúng (embedded system)
- Hệ thống thời gian thực (real-time)
  - Công nghệ đối tượng cho phép các hệ thống thời gian thực có thể phát triển với chất lượng cao hơn và linh hoạt hơn
    - Hệ thống vệ tinh
    - Các hệ thống quốc phòng và hàng không vũ trụ
    - ...





## 2.3. Các nguyên lý cơ bản của OO

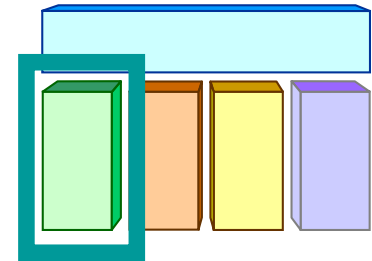
---



## a. Trừu tượng hóa (Abstraction)

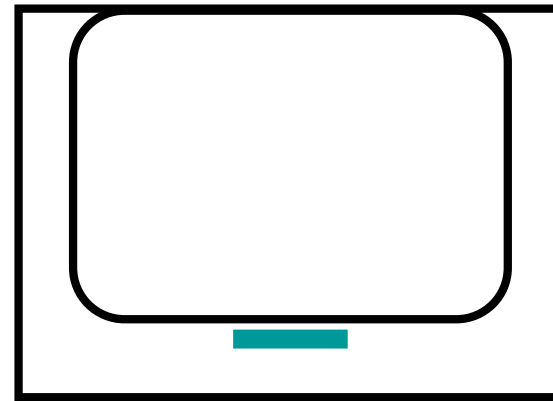
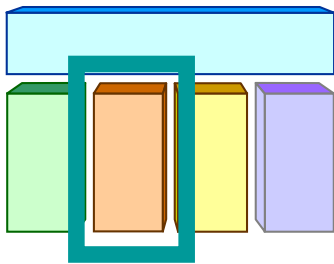
---

- Là quá trình loại bỏ đi các thông tin cụ thể và giữ lại những thông tin 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.



## b. Đóng gói (Encapsulation)

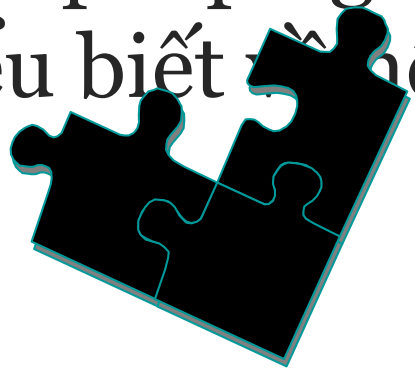
- Che giấu, ẩn đi chi tiết thực hiện bên trong
  - Cung cấp cho thế giới bên ngoài một giao diện
  - Người dùng không phụ thuộc vào việc sửa đổi sự thực thi bên trong



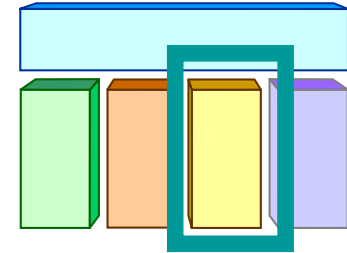
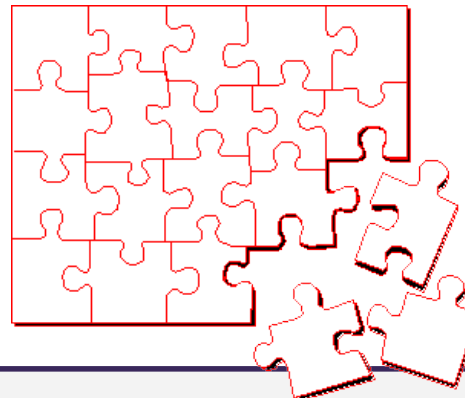
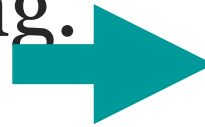
Tăng cường tính mềm dẻo

## c. Mô đun hóa (Modularity)

- Chia nhỏ hệ thống phức tạp thành những thành phần nhỏ có thể quản lý được.
- Cho phép người dùng hiểu biết về hệ thống.



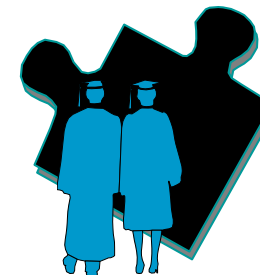
Hệ thống quản lý siêu thị sách



Hệ thống quản lý xuất nhập sách

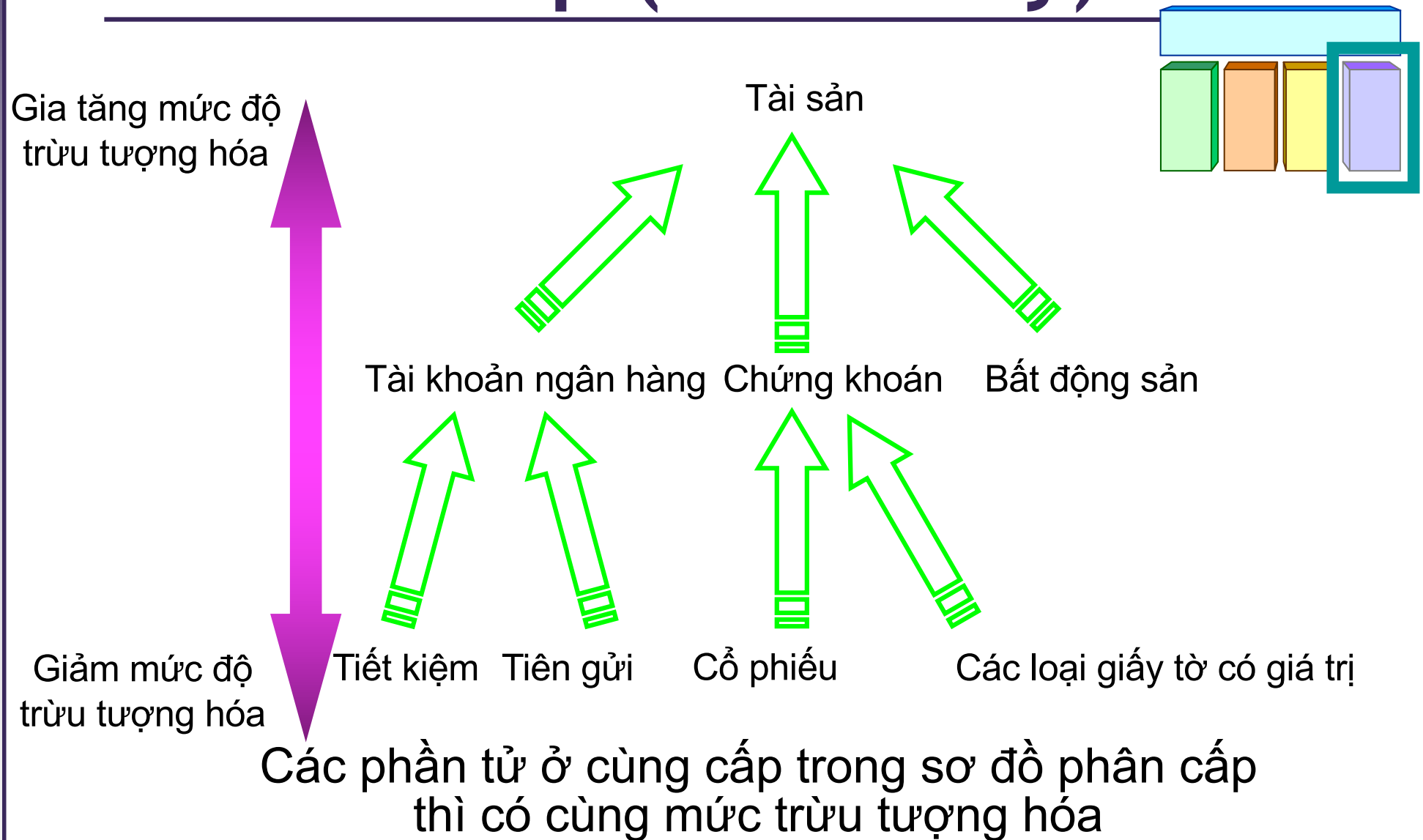


Hệ thống kế toán



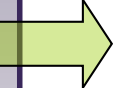
Hệ thống quản lý nhân viên

## d. Phân cấp (Hierarchy)



# Nội dung

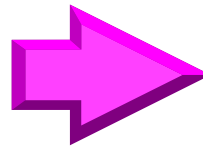
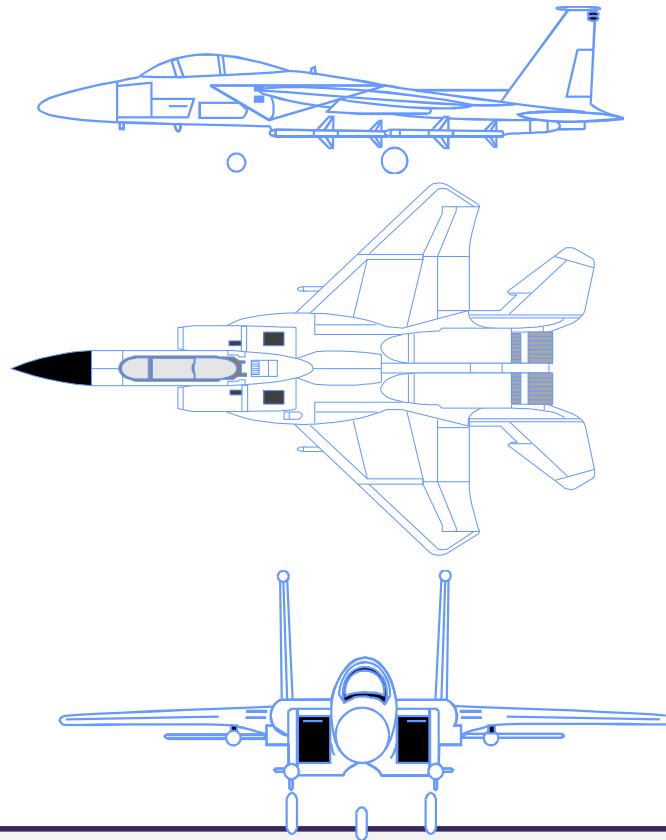
---

1. Lập trình hướng đối tượng
2. Công nghệ hướng đối tượng
-  3. Ngôn ngữ mô hình hóa thống nhất UML
4. Ngôn ngữ lập trình Java
5. Ví dụ và bài tập

## 3.1. Mô hình là gì?

---

- Mô hình là sự đơn giản hóa các vật thể, các đối tượng trong thế giới thực.

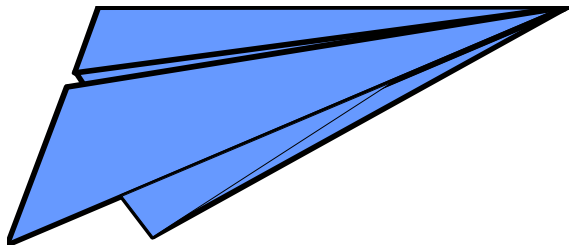


# Sự quan trọng của mô hình hóa

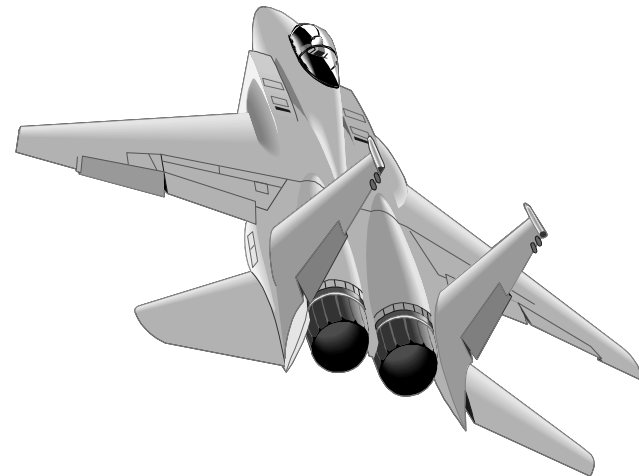
---

Mức độ quan trọng thấp

Mức độ quan trọng cao hơn



Máy bay giấy



Máy bay phản lực



## 3.2. Ngôn ngữ mô hình hóa thống nhất UML

---

- UML là ngôn ngữ để:
  - trực quan hóa (Visualizing)
  - xác định rõ (Đặc tả - Specifying)
  - xây dựng (Constructing)
  - tài liệu hóa (Documenting)

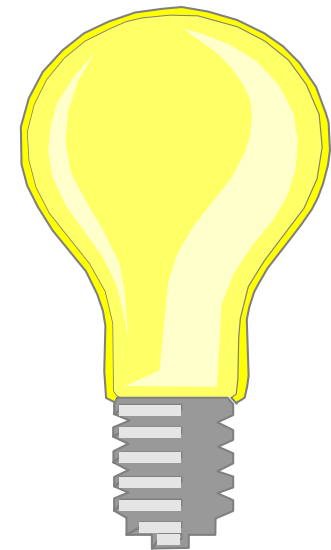
các cấu phần (artifact) của một hệ thống phần mềm.



# UML là ngôn ngữ trực quan

---

- UML là ngôn ngữ thống nhất trực quan giúp công việc được xử lý nhất quán, giảm thiểu lỗi xảy ra
  - Có những thứ mà nếu không mô hình hóa thì không hoặc khó có thể hiểu được
  - Mô hình trợ giúp hiệu quả trong việc liên lạc, trao đổi
    - Trong tổ chức
    - Bên ngoài tổ chức



# UML là ngôn ngữ để đặc tả

---

- UML xây dựng các mô hình chính xác, rõ ràng và đầy đủ.

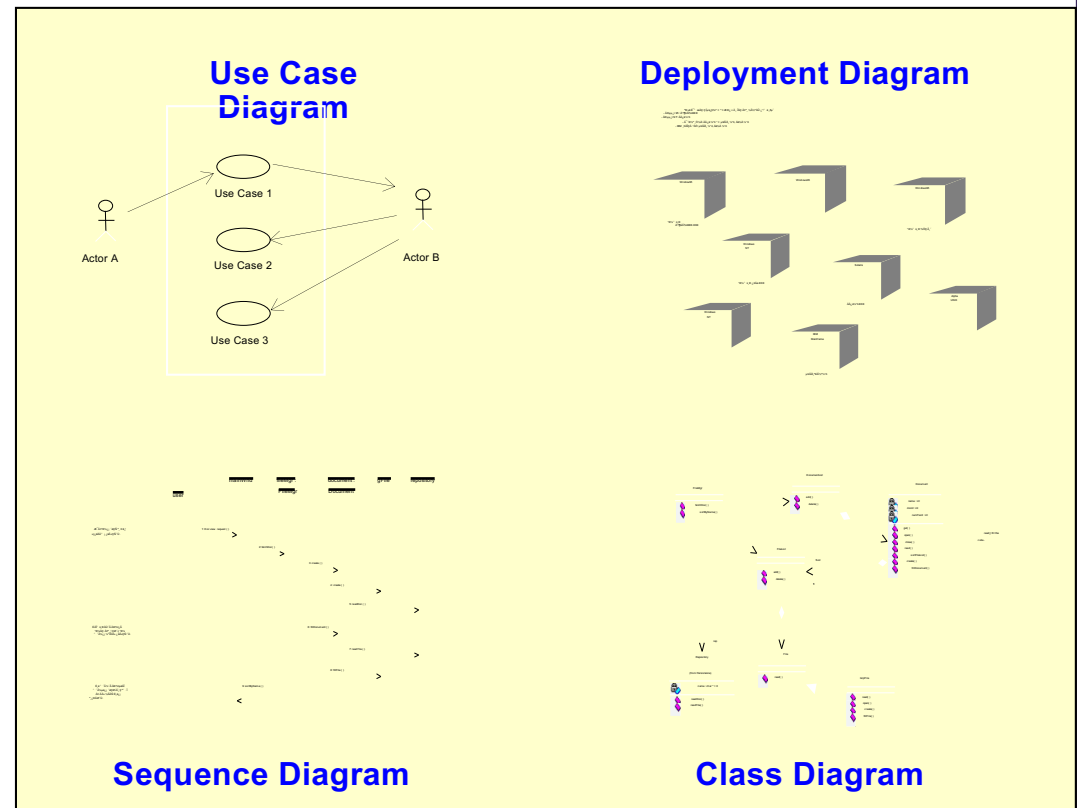


# UML là ngôn ngữ để xây dựng HT

- Các mô hình UML có thể kết nối trực tiếp với rất nhiều ngôn ngữ lập trình.
  - Ánh xạ sang Java, C++, Visual Basic...
  - Các bảng trong RDBMS hoặc kho lưu trữ trong OODBMS
  - Cho phép các kỹ nghệ xuôi (chuyển UML thành mã nguồn)
  - Cho phép kỹ nghệ ngược (xây dựng mô hình hệ thống từ mã nguồn)

# UML là ngôn ngữ để tài liệu hóa

- UML giúp tài liệu hóa về kiến trúc, yêu cầu, kiểm thử, lập kế hoạch dự án, và quản lý việc bàn giao PM
- Các biểu đồ khác nhau, các ghi chú, ràng buộc được đặc tả trong tài liệu



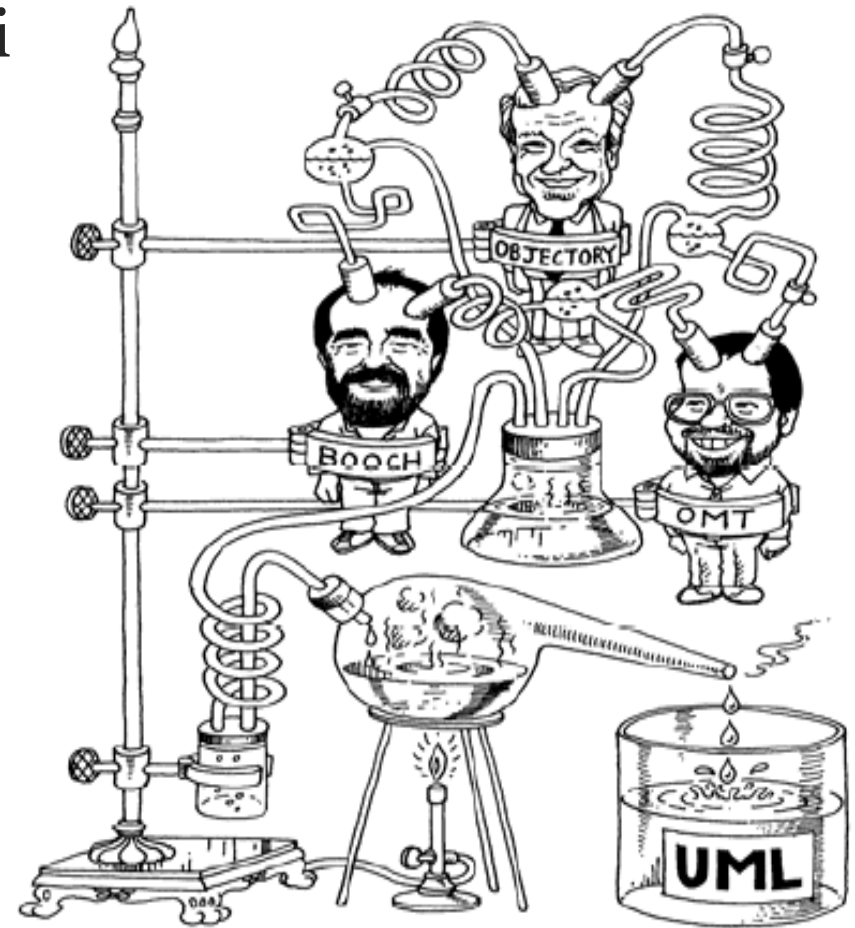
## 3.2.1. Lịch sử phát triển của UML

---

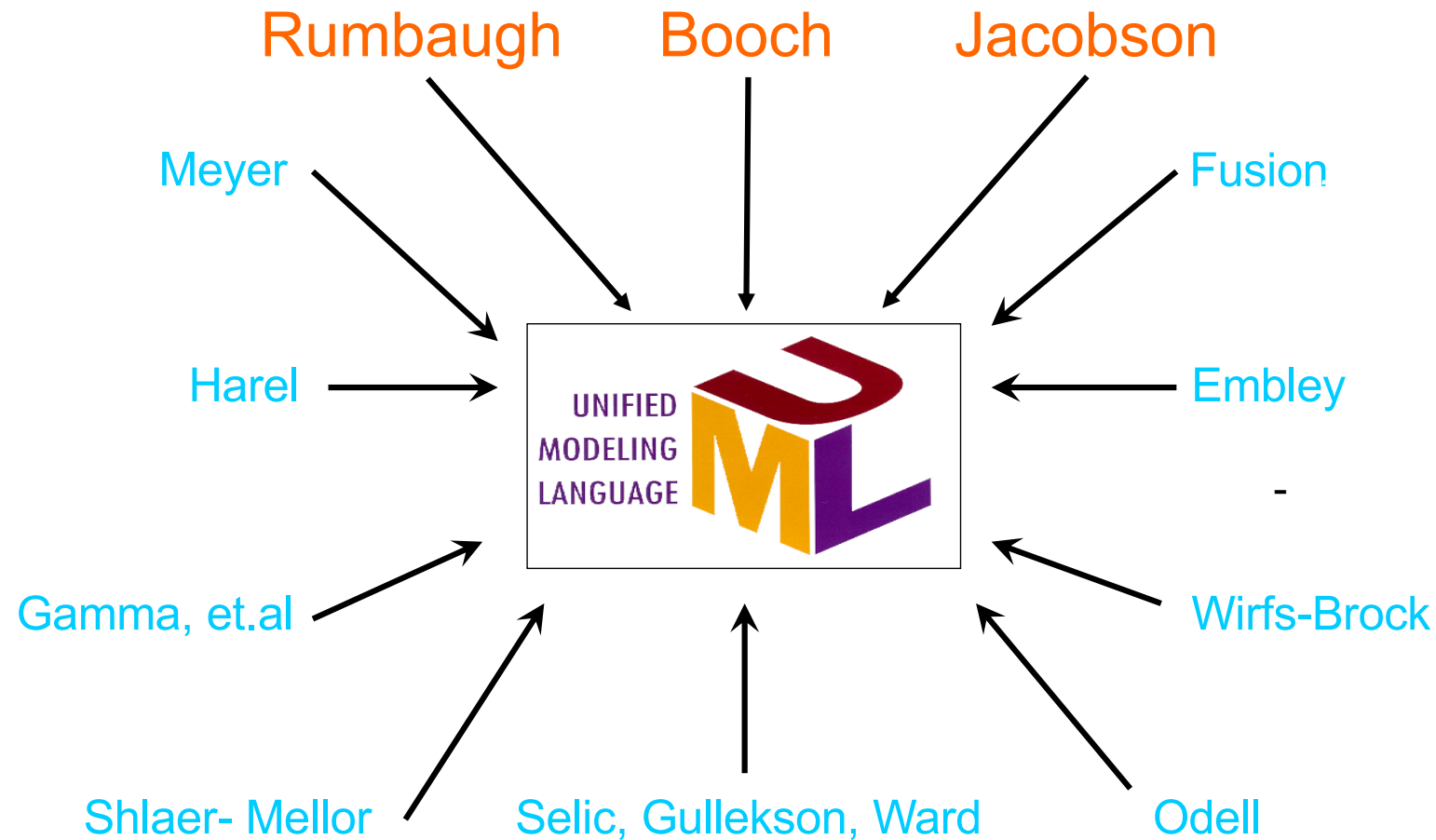
- Vào 1994, có hơn 50 phương pháp mô hình hóa hướng đối tượng:
    - Fusion, Shlaer-Mellor, ROOM, Class-Relation, Wirfs-Brock, Coad-Yourdon, MOSES, Syntropy, BOOM, OOSD, OSA, BON, Catalysis, COMMA, HOOD, Ooram, DOORS ...
  - “Meta-models” tương đồng với nhau
  - Các ký pháp đồ họa khác nhau
  - Quy trình khác nhau hoặc không rõ ràng
- Cần chuẩn hóa và thống nhất các phương pháp

## 3.2.1. Lịch sử phát triển của UML (2)

- UML được 3 chuyên gia hướng đối tượng hợp nhất các kỹ thuật của họ vào năm 1994:
  - Booch91 (Grady Booch): Conception, Architecture
  - OOSE (Ivar Jacobson): Use cases
  - OMT (Jim Rumbaugh): Analysis
- Thiết lập một phương thức thống nhất để xây dựng và “vẽ” ra các yêu cầu và thiết kế hướng đối tượng trong quá trình PTTK phần mềm → UML được công nhận là chuẩn chung vào năm 1997.

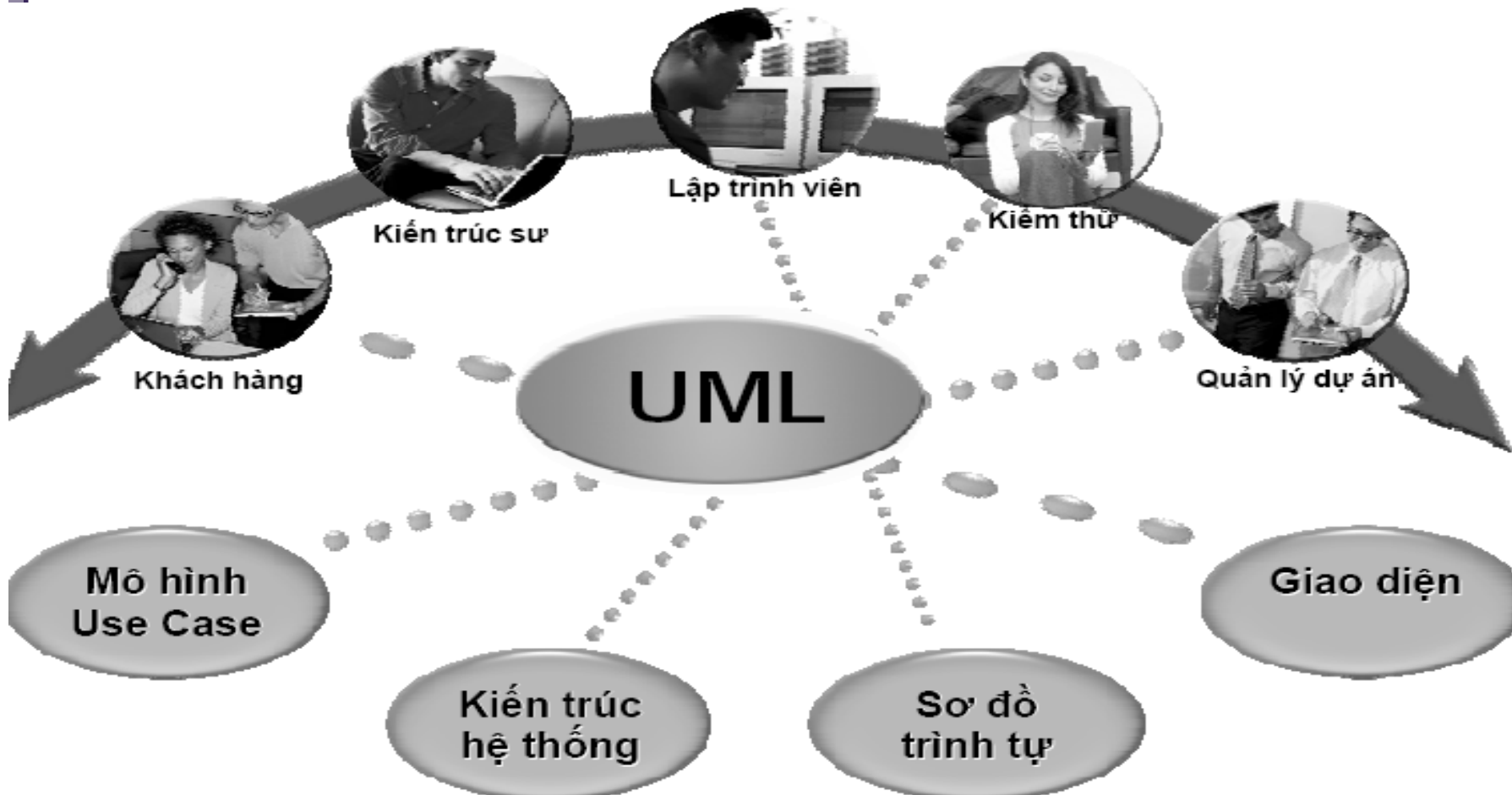


# UML là một ngôn ngữ hợp nhất

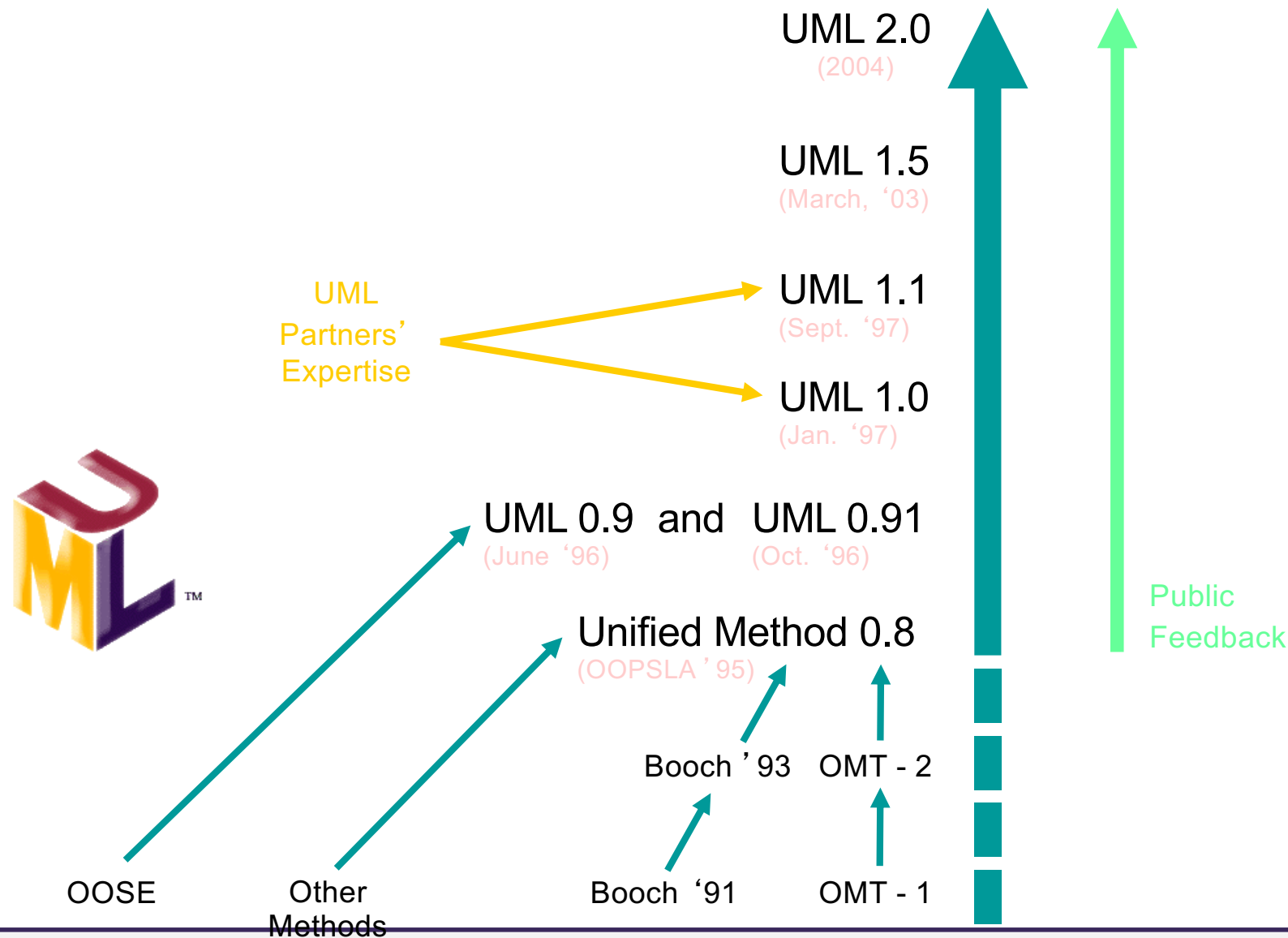




# UML là một ngôn ngữ thống nhất

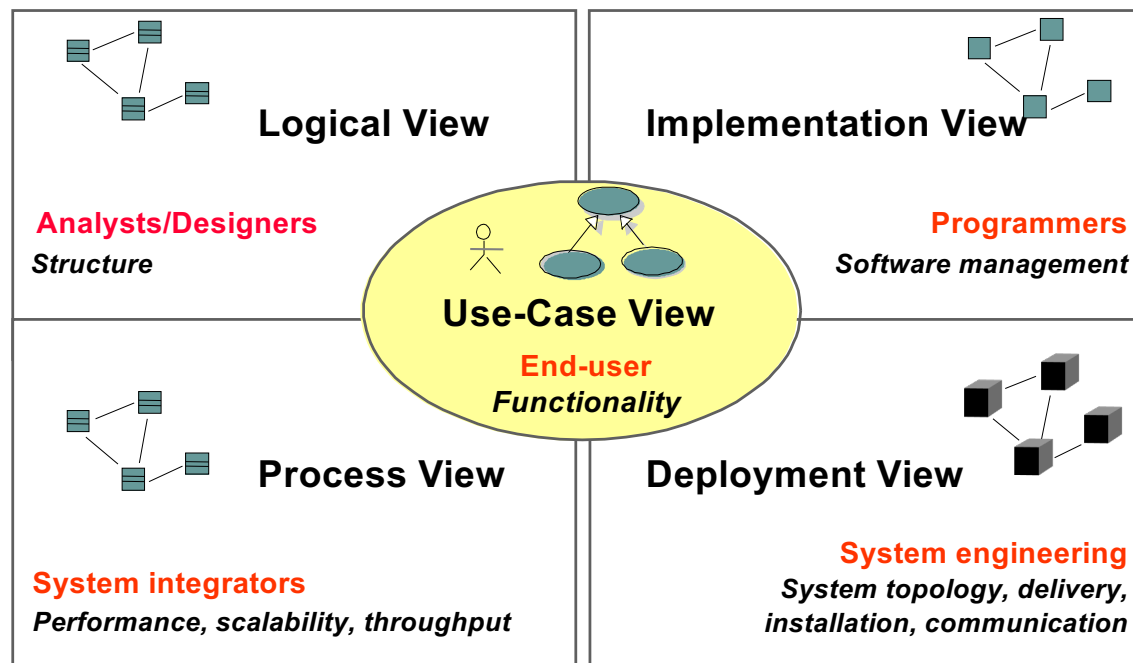


# 3.2.1. Lịch sử phát triển của UML (3)



## 3.2.2. Các khung nhìn của UML

- Khung nhìn của mô hình có ý nghĩa với những người tham gia nào đó
- 4 + 1 Architectural View



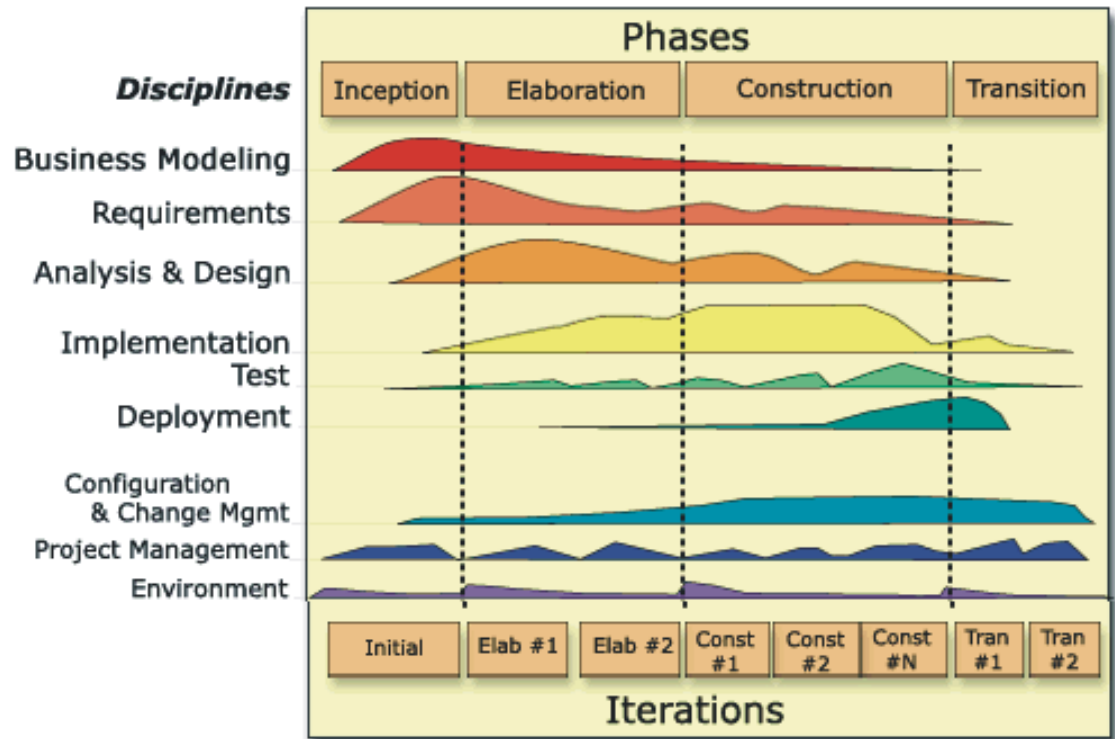
# Các biểu đồ UML

---

- Biểu đồ use case (Use Case Diagram)
- Biểu đồ tương tác (Interaction Diagrams)
  - Biểu đồ trình tự (Sequence Diagram)
  - Biểu đồ giao tiếp/cộng tác (Communication/Collaboration Diagram)
- Biểu đồ trạng thái (Statechart Diagram)
- Biểu đồ cấu trúc tĩnh (Static Structure Diagrams)
  - Biểu đồ lớp (Class Diagram)
  - Biểu đồ đối tượng (Object Diagram)
- Biểu đồ hoạt động (Activity Diagram)
- Biểu đồ thực thi (Implementation Diagrams)
  - Biểu đồ thành phần (Component Diagram)
  - Biểu đồ triển khai (Deployment Diagram)

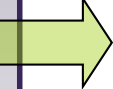
# Quy trình và UML

- UML là ký pháp chứ không phải là phương pháp
  - UML có thể áp dụng cho tất cả các pha của quy trình phát triển phần mềm
  - "Rational Unified Process" - quy trình phát triển cho UML



# Nội dung

---

1. Lập trình hướng đối tượng
2. Công nghệ hướng đối tượng
3. Ngôn ngữ mô hình hóa thống nhất UML
-  4. Ngôn ngữ lập trình Java
5. Ví dụ và bài tập

## 4.1. Java là gì?

- Ngôn ngữ LT HĐT
  - 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...
  - Bắt đầu được sử dụng từ năm 1995



ORACLE



Green Team and James Gosling  
(the leader)

## 4.1. Java là gì?

---

- Ngày nay, nhắc đến Java, không còn nhắc đến như một ngôn ngữ mà còn là một công nghệ, một nền tảng phát triển.
- Java có một cộng đồng phát triển mạnh mẽ
  - Một tập hợp các thư viện với số lượng lớn (từ Sun và các nguồn khác)



# J2SE (Java 2 Platform Standard Edition)

---

- <http://www.oracle.com/technetwork/java/javase/index-jsp-135232.html>
- Java 2 Runtime Environment, Standard Edition (J2RE):
  - Môi trường thực thi hay JRE cung cấp các Java API, máy ảo Java (JVM) và các thành phần cần thiết khác để chạy các applet và các ứng dụng viết bằng Java.
- Java 2 Software Development Kit, Standard Edition (J2SDK)
  - Tập mẹ của JRE, và chứa mọi thứ nằm trong JRE, bổ sung thêm các công cụ như là trình biên dịch và các trình gỡ lỗi cần để phát triển applet và các ứng dụng.

# J2EE (Java 2 Platform Enterprise Edition)

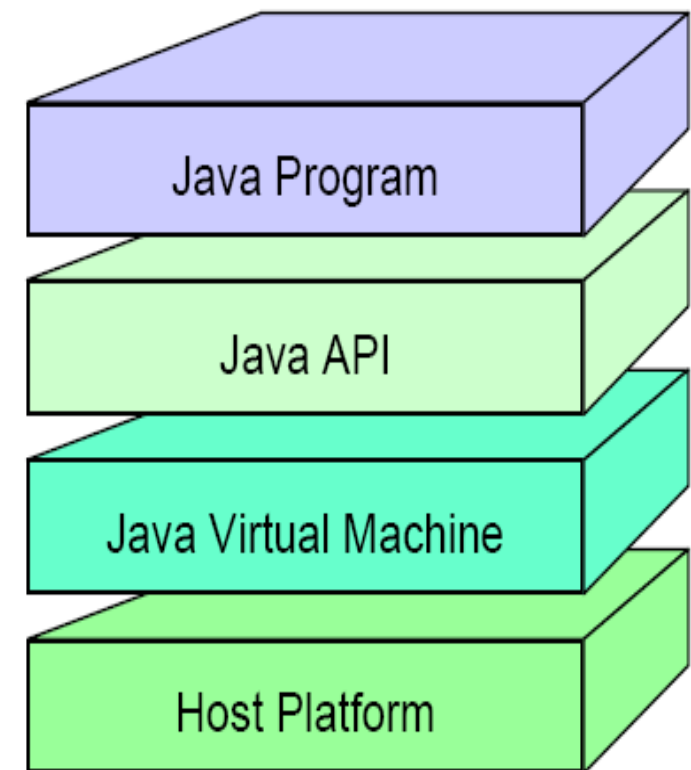
---

- <http://www.oracle.com/technetwork/java/javaee/overview/index.html>
- Service-Oriented Architecture (SOA) và Web services
- Các ứng dụng Web
  - Servlet/JSP
  - JSF...
- Các ứng dụng doanh nghiệp
  - EJB
  - JavaMail...
- ...

## 4.2. Nền tảng của Java (Java platform)

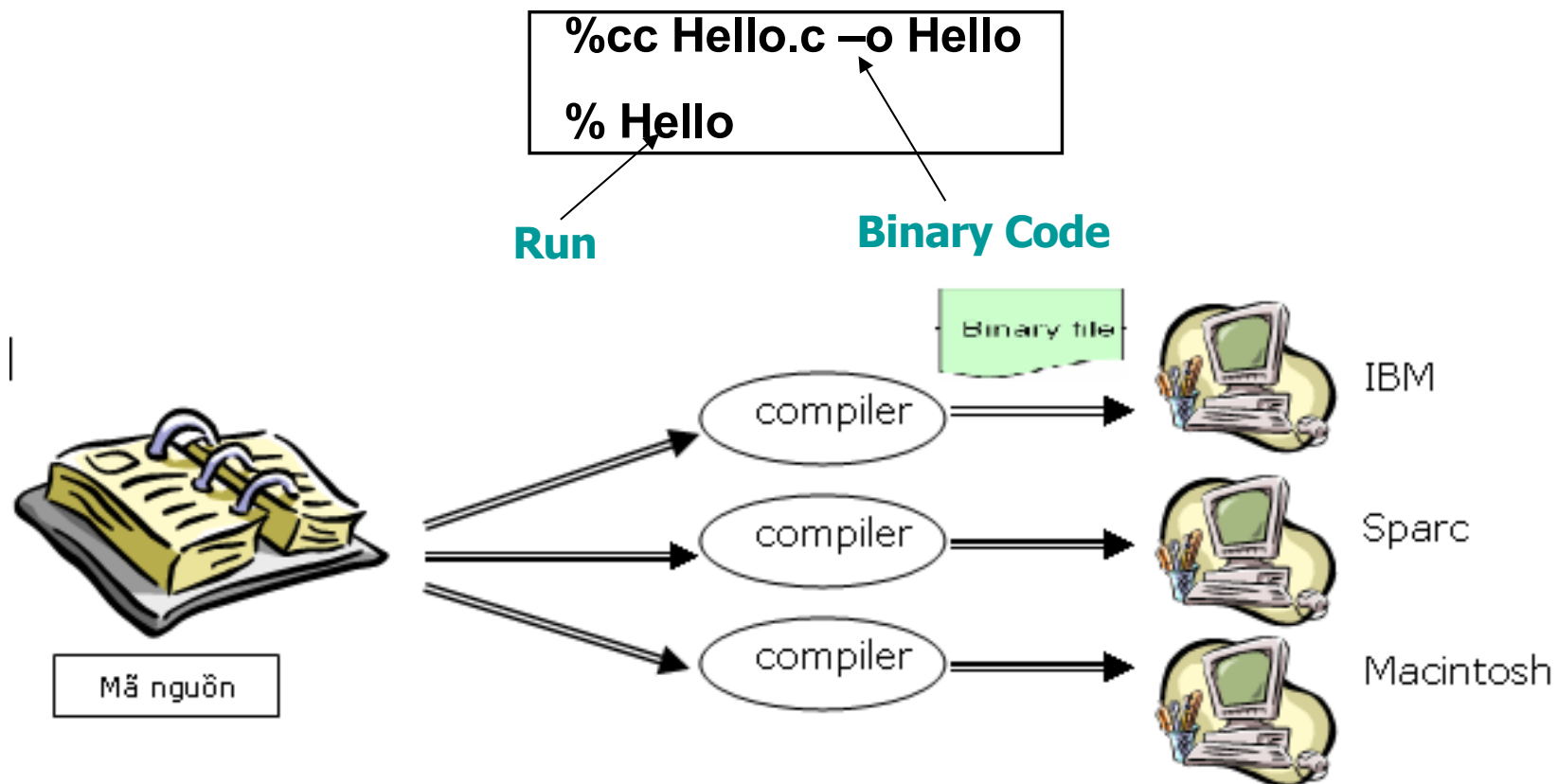
---

- Platform là môi trường phát triển hoặc triển khai.
- Java platform có thể chạy trên mọi hệ điều hành
  - Các platform khác phụ thuộc vào phần cứng
  - Java platform cung cấp:
    - Máy ảo Java - Java Virtual Machine (JVM).
    - Giao diện lập trình ứng dụng - Application Programming Interface (API).



## 4.3. Mô hình dịch của Java

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



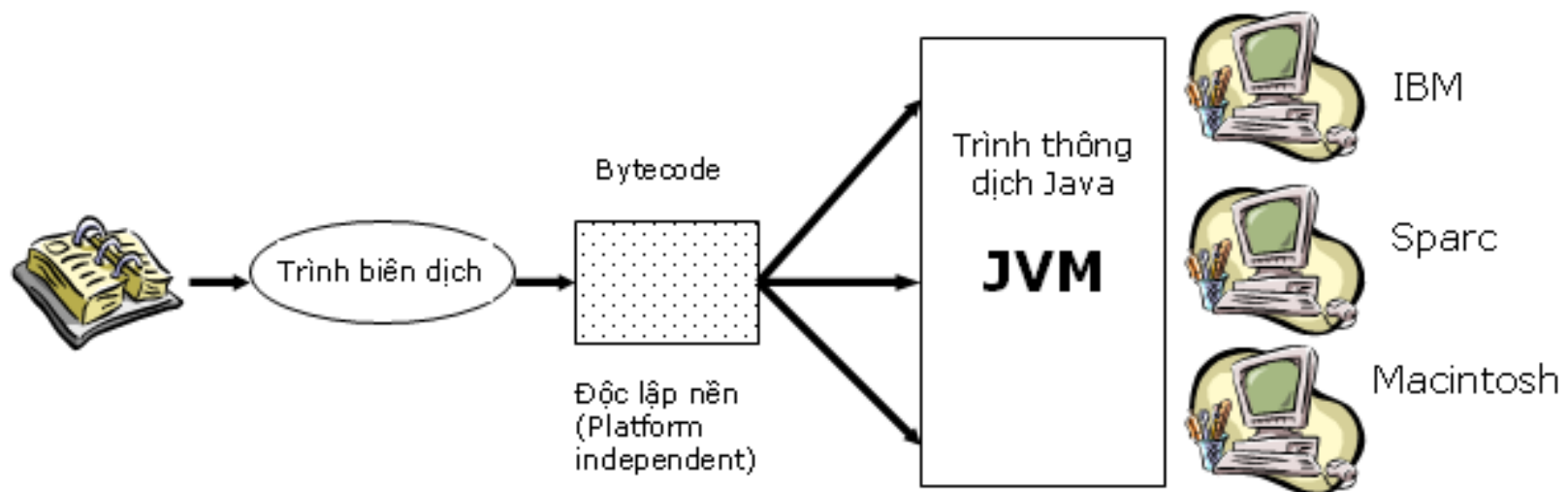
## 4.3. Mô hình dịch của Java (2)

- b. Mô hình dịch của Java:

- Mã nguồn được biên dịch thành bytecode rồi được thông dịch bởi JVM.

```
%javac Hello.java
Hello.class created
% java Hello
```

Run JVM Byte Code



## 4.3. Mô hình dịch của Java (3)

---

- Máy ảo Java (Java Virtual Machine):
  - Máy ảo Java là trái tim của ngôn ngữ Java
    - Đem đến cho các chương trình Java khả năng viết một lần nhưng chạy được ở mọi nơi
  - 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.

## 4.4. Các tính năng của Java

---

- Java được thiết kế:
  - Ngôn ngữ lập trình mạnh, đầy đủ tính năng và thuần hướng đối tượng.
  - Dễ học, cú pháp tương tự như C++
  - Độc lập nền tảng
  - Hỗ trợ phát triển các ứng dụng trong môi trường mạng
  - Lý tưởng cho các ứng dụng Web

## 4.4. Các tính năng của Java (2)

---

- **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.
- **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



## 4.4. Các tính năng của Java (3)

---

- 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
- Network capable
  - Java hỗ trợ phát triển các ứng dụng phân tán
  - Một số loại ứng dụng của Java được thiết kế để được truy cập thông qua trình duyệt Web.

## 4.4. Các tính năng của Java (3)

- Java có 50 từ khóa
  - assert (New in 1.5) enum (New in 1.5)

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

## 4.4. Các tính năng của Java (5)

---

- 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)

## 4.4. Các tính năng của Java (6)

---

- Các môi trường phát triển
  - Java Development Kit
    - Miễn phí trên Sun Website: [java.sun.com](http://java.sun.com)
    - Bao gồm: Trình biên dịch, JVM và các lớp đã có
  - Integrated Development Environments (IDEs):  
Cung cấp:
    - Các trình soạn thảo phức tạp
    - Các công cụ gỡ lỗi
    - Các công cụ phát triển đồ họa

## 4.5. Các kiểu chương trình Java

---

- Ứng dụng (Application)
  - Không cần chạy trên các trình duyệt
  - Có thể gọi các chức năng thông qua dòng lệnh hoặc menu lựa chọn (đồ họa)
- Applet
  - Chương trình đồ họa chạy trên trình duyệt tại máy trạm (client).
  - Có thể được xem bằng appletviewer hoặc nhúng trong trình duyệt Web có cài JVM.

## 4.5. Các kiểu chương trình Java (2)

---

- Ứng dụng Web (Web application)
  - Tạo ra các nội dung động trên server thay cho trên trình duyệt.
  - Chạy trong các ứng dụng server
  - Servlet: Kiểm soát các yêu cầu từ trình duyệt và trả lại các phản hồi
  - JavaServer Page (JSP): Các trang HTML được nhúng với mã Java.

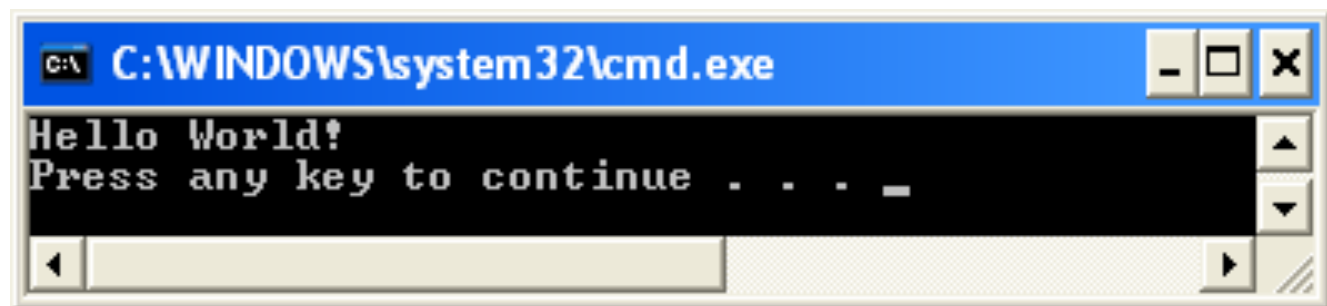
# Nội dung

---

1. Lập trình hướng đối tượng
2. Công nghệ hướng đối tượng
3. Ngôn ngữ mô hình hóa thống nhất UML
4. Ngôn ngữ lập trình Java
5. Ví dụ và bài tập

# Ví dụ 1 - HelloWorld

```
//Trong file HelloWorld.java
//Chương trình hiển thị dòng chữ "Hello World"
public class HelloWorld {
    /* Phương thức main sẽ được gọi đầu tiên
    trong bất kỳ ứng dụng Java nào */
    public static void main(String args[]){
        System.out.println("Hello World!");
    } // kết thúc phương thức main
} // kết thúc lớp HelloWorld
```





# Ví dụ 1 (tiếp)

---

- Chú thích (Comment)
  - Trên 1 dòng: Bắt đầu bằng: //
  - Nhiều dòng: /\* ... \*/
- Java phân biệt chữ hoa chữ thường
- Từ khóa có sẵn của Java:
  - class: Khai báo lớp
  - public: Quy định phạm vi truy cập
- Tên lớp chứa hàm main phải trùng với tên file .java.

# Cài đặt và chạy thử chương trình Java

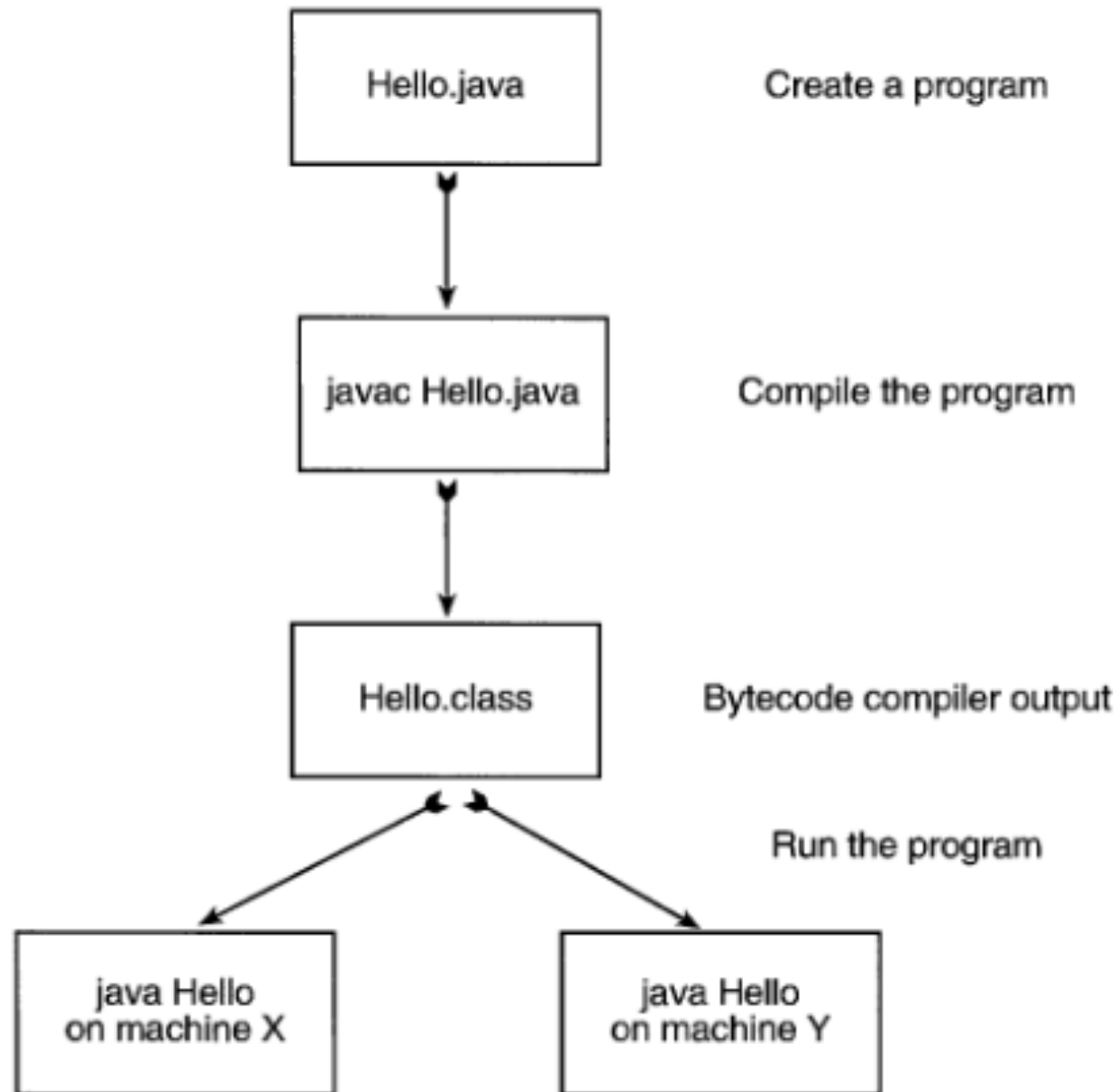
---

- Bước 1: Cài đặt j2sdk1.5/6/7, cài đặt các biến môi trường (nếu dùng cmd)
- Bước 2: Cài trình soạn thảo TextPad/JCreator/NetBean/Eclipse...
- Bước 3: Lập trình/Viết mã nguồn
- Bước 4: Dịch
  - cmd: `javac HelloWorld.java`
  - Textpad: Ctrl + 1
  - JCreator: F7 hoặc Build → Build Project/File
- Bước 5: Chạy chương trình
  - cmd: `java HelloWorld.class`
  - Textpad: Ctrl + 2
  - JCreator: F5 hoặc Run → Run Project/File

# Biến môi trường

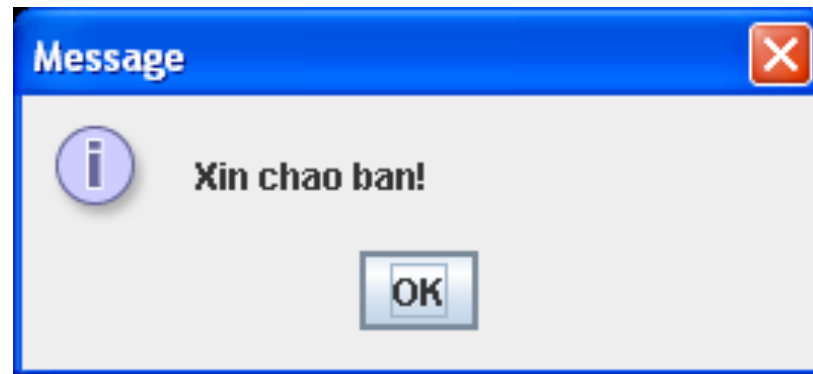
---

- PATH = ...;C:\Program Files\Java\jdk1.6\bin
- CLASSPATH = C:\Program Files\Java\jdk1.6\lib;.;C:\Program Files\Java\jdk1.6\include



## Ví dụ 2 - GUI

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



# Ví dụ 3 - Nhập, xuất dữ liệu

```
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);  
    }  
}
```

