





Bộ môn Công nghệ Phần mềm
Viện CNTT & TT
Trường Đại học Bách Khoa Hà Nội

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài 01. Tổng quan về OOP


1



Nội dung

1. Lập trình hướng đối tượng
2. Công nghệ hướng đối tượng
3. Ngôn ngữ lập trình Java
4. Ví dụ và bài tập


2



Nội dung

1. **Lập trình hướng đối tượng**
2. Công nghệ hướng đối tượng
3. Ngôn ngữ lập trình Java
4. Ví dụ và bài tập

3



1.1. Lịch sử phát triển của các NNLT

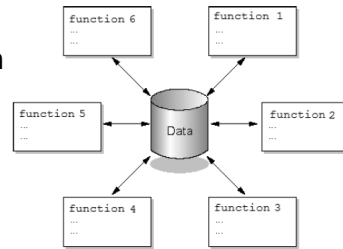
- a. Hợp ngữ (Assembly language):
 - Là một ngôn ngữ lập trình tuần tự, gần với tập các lệnh mã máy của CPU.
 - Khó nhớ, khó viết, nhất là với những bài toán phức tạp.
 - Khó sửa lỗi, bảo trì.

4

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

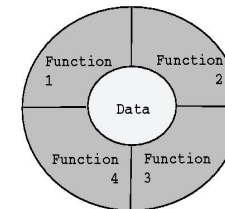


5

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 đối tượng.



6

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,...



7

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.

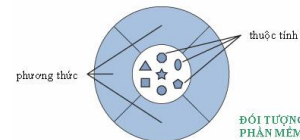
- Tất cả có trạng thái (state) và hành động (behaviour)

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

8

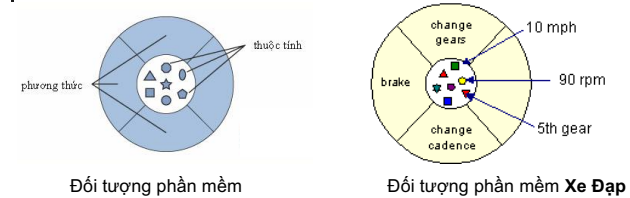
Đối tượng phần mềm

- 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)



9

Đối tượng

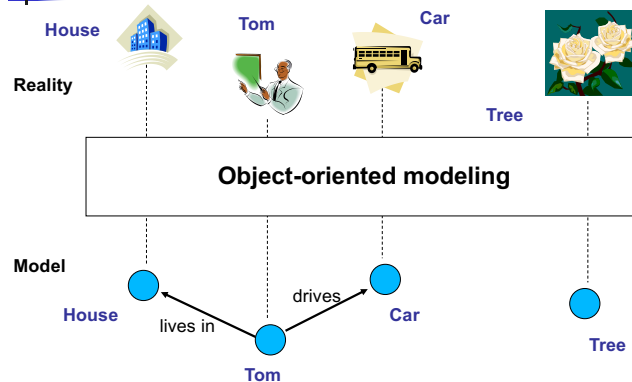


Đối tượng (object) là một thực thể phần mềm bao bọc các **thuộc tính** và các **phương thức** liên quan.

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**.

10

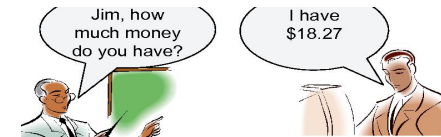
Mô hình hóa đối tượng



11

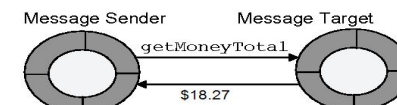
Tương tác giữa các đối tượng

- 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)



12

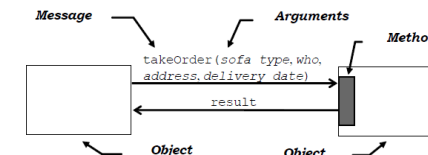
Gọi hàm vs. Gửi thông điệp

- **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.

13

Thông điệp vs. Phương thức

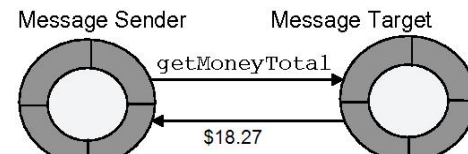
- **Thông điệp**
 - Được gửi từ đối tượng này đến đối tượng kia, không bao gồm đoạn mã thực sự sẽ được thực thi
- **Phương thức**
 - Thủ tục/hàm trong ngôn ngữ lập trình cấu trúc
 - Là sự thực thi dịch vụ được yêu cầu bởi thông điệp
 - Là đoạn mã sẽ được thực thi để đáp ứng thông điệp được gửi đến cho đối tượng



14

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)



15

1.3. Lớp đối tượng

- Trong thế giới thực có nhiều đối tượng cùng loại.
- Chương trình hướng đối tượng có nhiều đối tượng cùng loại chia sẻ những đặc điểm chung.
- Ví dụ



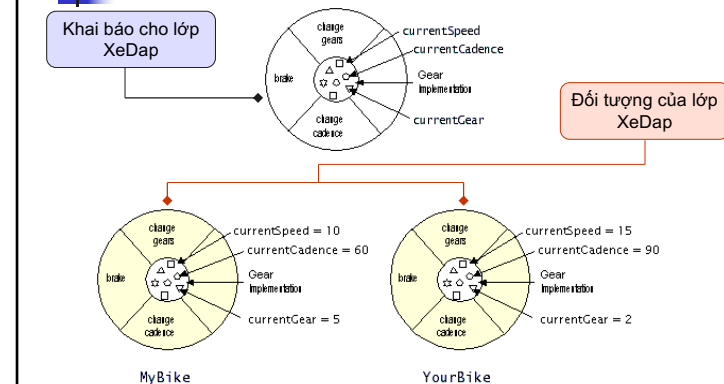
16

Lớp

- 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.

17

Ví dụ Lớp Xe đạp



18

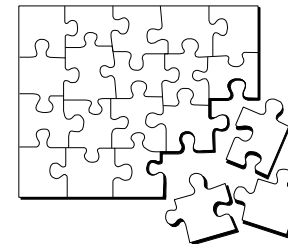
Nội dung

1. Lập trình hướng đối tượng
2. **Công nghệ hướng đối tượng**
3. Ngôn ngữ lập trình Java
4. Ví dụ và bài tập

19

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)

20

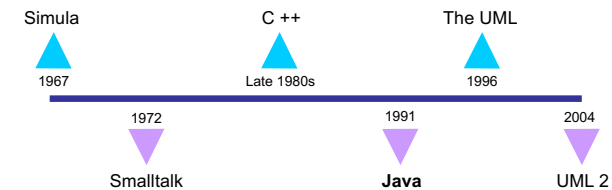
Đặc tính của LT HĐT

- 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

21

2.1. Lịch sử phát triển của OT

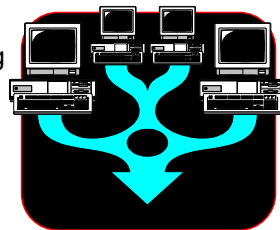
- Các mốc chính của công nghệ đối tượng



22

2.2. OT được sử dụng ở đâu?

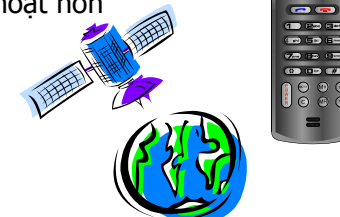
- Các hệ thống Client/Server và phát triển Web
 - Công nghệ đối tượng cho phép các công ty đóng gói thông tin doanh nghiệp trong các đối tượng và giúp phân phối quá trình xử lý qua mạng Internet hoặc một mạng máy tính.



23

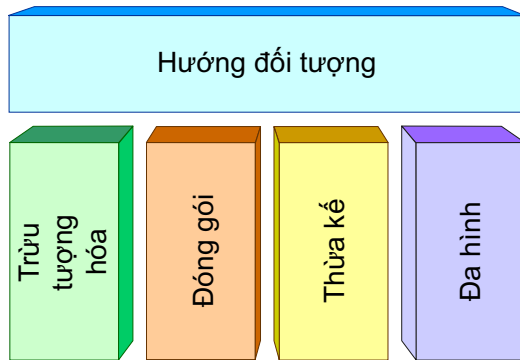
2.2. OO được sử dụng ở đâu? (2)

- 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
 - ...



24

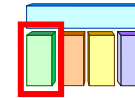
2.3. Các nguyên lý cơ bản của OO



25

a. 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.



26

Ví dụ: Trừu tượng hóa



27

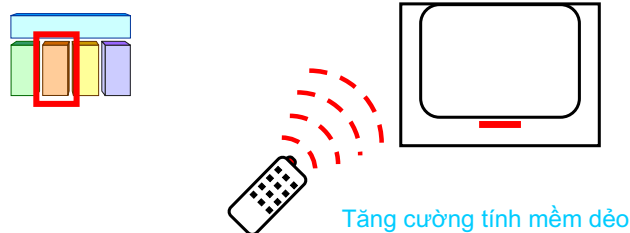
b. Đóng gói (Encapsulation)



28

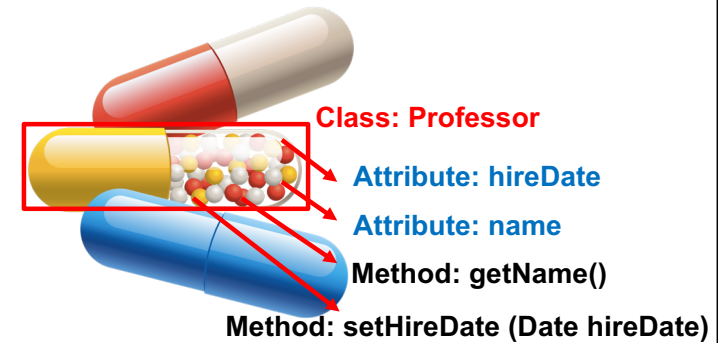
b. Đóng gói (Encapsulation)

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



29

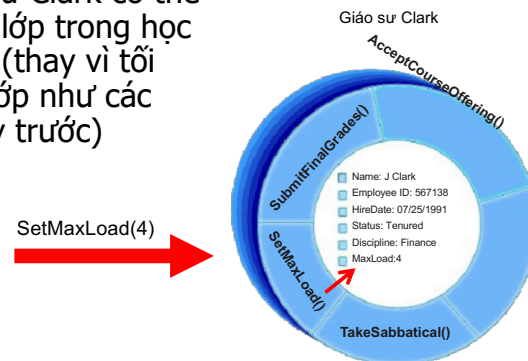
b. Đóng gói (Encapsulation)



30

Minh họa việc đóng gói

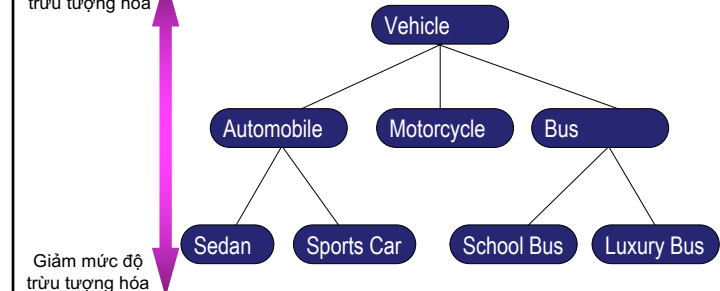
- Giáo sư Clark có thể dạy 4 lớp trong học kỳ tới (thay vì tối đa 3 lớp như các học kỳ trước)



31

c. Thừa kế (Inheritance)

Gia tăng mức độ trừu tượng hóa



Giảm mức độ trừu tượng hóa

35

d. Đa hình (polymorphism)

- Đa hình: "one name, many forms"
- Nạp chồng phương thức (Method overloading)
 - 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

36

Nội dung

1. Lập trình hướng đối tượng
2. Công nghệ hướng đối tượng
3. **Ngôn ngữ lập trình Java**
4. Ví dụ và bài tập

37

3.1. Java là gì?

- Java là một ngôn ngữ lập trình HĐT được phát triển bởi Sun Microsystems.
- Java là một ngôn ngữ lập trình khá trẻ
 - 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



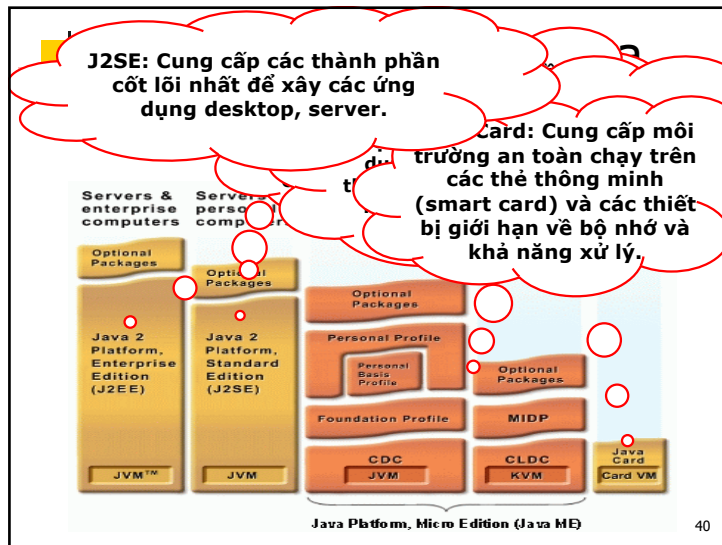
Green Team and James Gosling
(the leader)

38

3.1. 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)

39



J2SE (Java 2 Platform Standard Edition)

- <http://java.sun.com/j2se>
- 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.

41

J2EE (Java 2 Platform Enterprise Edition)

- <http://java.sun.com/j2ee>
- Service-Oriented Architecture (SOA) và Web services
- Các ứng dụng Web
 - Servlet/JSP
 - JSF...
- Các ứng dụng doanh nghiệp
 - EJB
 - JavaMail...
- ...

43

Lịch sử phát triển của J2SE

- JDK 1.1.4 (Sparkler): 12 tháng 9, 1997
- JDK 1.1.5 (Pumpkin): 3 tháng 12, 1997
- JDK 1.1.6 (Abigail): 24 tháng 4, 1998
- JDK 1.1.7 (Brutus): 28 tháng 9, 1998
- JDK 1.1.8 (Chelsea): 8 tháng 4, 1999
- J2SE 1.2 (Playground): 4 tháng 12, 1998
- J2SE 1.2.1 (none): 30 tháng 3, 1999
- J2SE 1.2.2 (Cricket): 8 tháng 7, 1999
- J2SE 1.3 (Kestrel): 8 tháng 5, 2000
- J2SE 1.3.1 (Ladybird): 17 tháng 5, 2001

44

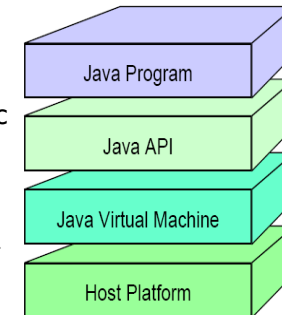
Lịch sử phát triển của J2SE (2)

- J2SE 1.4.0 (Merlin) 13 tháng 2, 2002
- J2SE 1.4.1 (Hopper) 16 tháng 9, 2002
- J2SE 1.4.2 (Mantis) 26 tháng 6, 2003
- J2SE 5 (1.5.0) (Tiger) 29 tháng 9, 2004
- **Java SE 6 (Mustang)**, 11 tháng 12, 2006
 - Các bản cập nhật 2 và 3 được đưa ra vào năm 2007
 - Bản cập nhật 4 đưa ra tháng 1 năm 2008.
- **Java SE 7 (Dolphin)**, 4/2008.

45

3.2. Nền tảng của Java (Java platform)

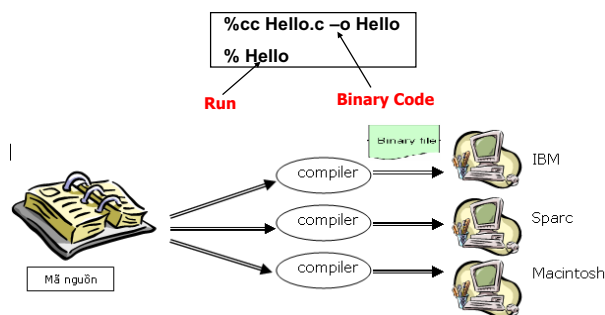
- Platform là môi trường phát triển hoặc triển khai.
- Java platform có thể chạy trên mọi hệ điều hành
 - Các platform khác phụ thuộc vào phần cứng
 - Java platform cung cấp:
 - Máy ảo Java - Java Virtual Machine (JVM).
 - Giao diện lập trình ứng dụng - Application Programming Interface (API).



46

3.3. Mô hình dịch của Java

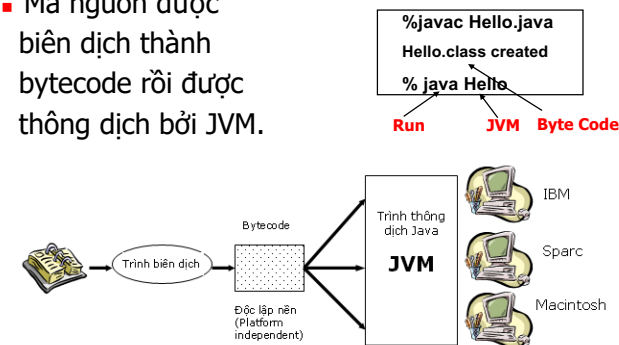
- a. Mô hình biên dịch truyền thống:
 - Mã nguồn được biên dịch thành mã nhị phân.



47

3.3. 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.



48

3.3. 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.

49

3.4. 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

50

3.4. 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

51

3.4. 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.

52

3.4. Các tính năng của Java (3)

■ Java có 50 từ khóa

- assert (New in 1.5) enum (New in 1.5)

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

53

3.4. Các tính năng của Java (4)

■ Bảo mật

- 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

54

3.4. 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)

55

3.4. Các tính năng của Java (6)

■ Các môi trường phát triển

- Java Development Kit
 - Miễn phí trên Sun Website: 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

56

3.5. Các kiểu chương trình Java

- **Ứng dụng (Application)**
 - Không cần chạy trên trình duyệt
 - Có thể gọi các chức năng thông qua dòng lệnh hoặc menu lựa chọn (đồ họa)
 - Phương thức main() là điểm bắt đầu thực hiện ứng dụng
- **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.

57

3.5. 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.

58

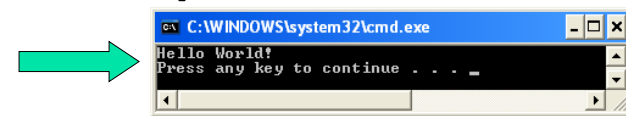
Nội dung

1. Lập trình hướng đối tượng
2. Công nghệ hướng đối tượng
3. Ngôn ngữ lập trình Java
4. **Ví dụ và bài tập**

59

Ví dụ 1 - HelloWorld

```
// 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
```



60

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.

61

Cài đặt và chạy thử chương trình Java

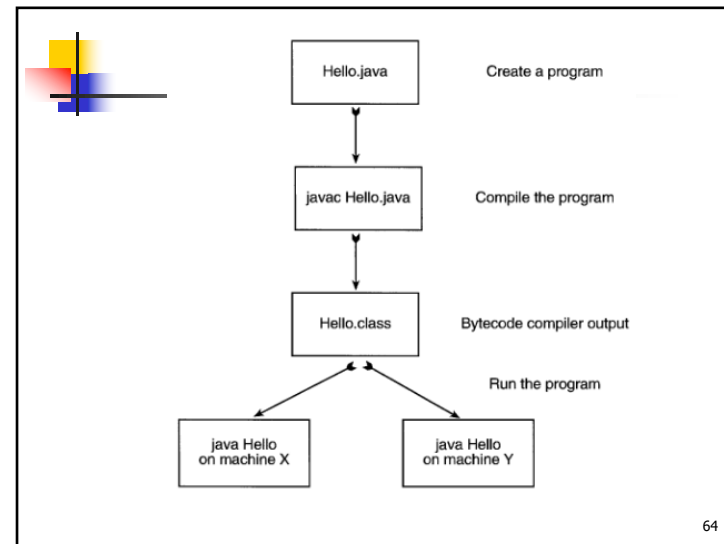
- Bước 1: Cài đặt j2sdk1.5/6/7, cài đặt các biến môi trường (nếu dùng cmd)
- Bước 2: Cài trình soạn thảo TextPad/JCreator/NetBean/Eclipse...
- Bước 3: Lập trình/Viết mã nguồn
- Bước 4: Dịch
 - cmd: javac HelloWorld.java
 - Textpad: Ctrl + 1
 - JCreator: F7 hoặc Build → Build Project/File
- Bước 5: Chạy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

62

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

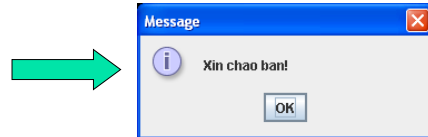
63



64

Ví dụ 2 - GUI

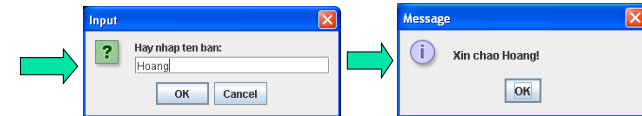
```
import javax.swing.JOptionPane;
public class FirstDialog{
    public static void main(String[] args){
        JOptionPane.showMessageDialog(null,
            "Xin chào bạn!");
        System.exit(0);
    }
}
```



65

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("Hãy nhập
            tên bạn:");
        JOptionPane.showMessageDialog(null,
            "Xin chào " + result + "!");
        System.exit(0);
    }
}
```



66

Ví dụ 4

```
import java.util.Scanner;
public class Demo {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Nhập i = ");
        int i = s.nextInt();
        System.out.println(i);

        System.out.print("Nhập l = ");
        long l = s.nextLong();
        System.out.println(l);

        s.close();
    }
}
```

67

Bài tập:

- 1. Viết chương trình nhập hai số nguyên và hiển thị tổng, hiệu, tích và thương của 2 số nguyên vừa nhập

68