

MỤC LỤC

DANH MỤC THUẬT NGỮ TIẾNG ANH.....	7
LỜI NÓI ĐẦU.....	8
Chương 1. TỔNG QUAN VỀ LẬP TRÌNH JAVA.....	10
1.1. Lịch sử ra đời và phát triển của Java	10
1.2. Đặc trưng ngôn ngữ Java	10
1.3. Các ứng dụng của Java.....	12
1.4. Dịch và thực thi một chương trình viết bằng Java	13
1.5. Kiến trúc chương trình xây dựng trên Java.....	13
1.5.1. Kiến trúc chương trình Java	13
1.5.2. Chương trình Java đầu tiên.....	16
Chương 2. CẤU TRÚC LẬP TRÌNH CƠ BẢN TRONG JAVA.....	20
2.1. Các kiểu dữ liệu	20
2.2. Biến	21
2.3. Toán tử	22
2.4. Strings và StringBuider.....	23
2.4.1. Lớp String.....	23
2.4.2. StringBuffer và StringBuilder	27
2.5. Nhập/Xuất dữ liệu.....	27
2.6. Câu lệnh và các cấu trúc lệnh trong Java.....	31
2.6.1. Lệnh, khối lệnh trong Java.	32
2.6.2. Câu lệnh if ...else	32
2.6.3. Câu lệnh switch-case	33
2.6.4. Vòng lặp While.....	35
2.6.5. Vòng lặp do-while	35
2.6.6. Vòng lặp for.....	36
2.6.7. Lệnh break và continue.....	37
2.7. Số lớn trong java	38
2.8. Mảng	38
2.8.1. Vòng lặp “for each”	39
2.8.2. Khởi tạo mảng và mảng nặc danh	40
2.8.3. Copy mảng.....	41
2.8.4. Các tham số dòng lệnh.....	41
2.8.5. Sắp xếp mảng.....	42

2.8.6.	Mảng nhiều chiều	42
Chương 3. ĐỐI TƯỢNG, LỚP, KẾ THỪA, GIAO DIỆN.....		49
3.1.	Đối tượng, lớp, lớp trừu tượng.....	49
3.1.1.	Khái niệm	49
3.1.2.	Khai báo lớp	49
3.1.3.	Thuộc tính của lớp	50
3.1.4.	Phương thức của lớp	52
3.1.5.	Chỉ định truy xuất lớp.....	55
3.2.	Tạo đối tượng.....	55
3.3.	Kế thừa và đa hình	57
3.3.1.	Mô tả kế thừa	57
3.3.2.	Kế thừa đơn	57
3.3.3.	Kế thừa kép.....	61
3.3.4.	Lớp cha, lớp con	63
3.3.5.	Cách sử dụng từ khóa super	64
3.3.6.	Cách nạp chồng phương thức	64
3.3.7.	Đa hình	66
3.4.	Lớp trừu tượng	68
3.5.	Lớp Object	71
3.5.1.	Phương thức equals	71
3.5.2.	Phương thức toString ().....	72
3.6.	Giao diện	72
3.6.1.	Mô tả giao diện.....	72
3.6.2.	Mục đích sử dụng giao diện	73
3.6.3.	So sánh giao diện và lớp trừu tượng.....	73
3.7.	Lớp nội.....	74
3.7.1.	Các kiểu lớp nội.....	75
3.8.	Xử lý ngoại lệ.....	77
3.8.1.	Giới thiệu	77
3.8.2.	Mục đích của việc xử lý ngoại lệ.....	77
3.8.3.	Lớp Exception	77
3.8.4.	Xử lý ngoại lệ	77
3.8.5.	Mô hình xử lý ngoại lệ	78
3.8.6.	Các khối chứa nhiều Catch	79

3.8.7.	Khởi ‘finally’	81
3.8.8.	Các ngoại lệ được định nghĩa với lệnh ‘throw’ và ‘throws’	83
3.8.9.	Danh sách các ngoại lệ	84
3.9.	Giao diện Collection	85
3.9.1.	Giới thiệu	85
3.9.2.	Các kiểu collection	85
3.9.3.	Collection frame work	91
3.10.	Thread	94
3.10.1.	Thread là gì, Tạo Thread	94
3.10.2.	Các trạng thái của Thread	94
3.10.3.	Các phương thức của lớp Thread	95
3.10.4.	Quản lý Thread	96
3.10.5.	Luồng chạy ngầm (daemon)	96
3.11.	Đa luồng và tương tranh	97
3.11.1.	Định nghĩa đa luồng	97
3.11.2.	Sử dụng phương thức isAlive() và join()	100
3.11.3.	Điều kiện tương tranh và cách khắc phục	101
3.11.4.	Cách sử dụng phương thức wait() và notify()	102
3.11.5.	Định nghĩa deadlock và khắc phục	103
CHƯƠNG 4. LẬP TRÌNH GIAO DIỆN VỚI JAVA SWING		111
4.1.	Giới thiệu Swing và mô hình MVC	111
4.1.1.	Lợi ích của sử dụng Swing so với AWT	112
4.1.2.	Mô tả container trong Swing	113
4.2.	Quản lý Layout	113
4.2.1.	BorderLayout	113
4.2.2.	GridLayout	115
4.2.3.	GirdBagLayout	116
4.2.4.	GroupLayout	116
4.3.	Text Input	116
4.3.1.	TextFields	116
4.3.2.	Label	117
4.3.3.	Password Fields	118
4.3.4.	TextAreas	118
4.3.5.	Scroll Panes	119

4.4.	Các thành phần lựa chọn.....	120
4.4.1.	Checkboxes.....	120
4.4.2.	Radio Button.....	120
4.4.3.	ComboBoxes	121
4.4.4.	Slider.....	123
4.5.	Menu	124
4.5.1.	Xây dựng menu	124
4.5.2.	Icons và Menu Items.....	124
4.5.3.	Pop-Up Menu	125
4.5.4.	Hiển thị và ẩn các mục trong menu	125
4.5.5.	Toolbars	125
4.5.6.	Tooltips	127
4.6.	Dialog Boxes.....	127
4.6.1.	Tạo Dialogs.....	127
4.6.2.	Option Dialogs.....	128
4.6.3.	Trao đổi dữ liệu	132
4.6.4.	File Dialogs.....	132
4.6.5.	Các lựa chọn màu	133
	Chương 5. LUỒNG VÀ TẬP TIN.....	139
5.1.	Mở đầu	139
5.2.	Luồng	139
5.2.1.	Khái niệm luồng	139
5.2.2.	Luồng byte (Byte Streams).....	139
5.2.3.	Luồng ký tự (Character Streams)	140
5.2.4.	Những luồng được định nghĩa trước (The Predefined Streams)	141
5.3.	Sử dụng luồng Byte.....	141
5.3.1.	Đọc dữ liệu từ Console	142
5.3.2.	Xuất dữ liệu ra Console	143
5.3.3.	Đọc và ghi file dùng luồng Byte.....	144
5.3.4.	Đọc và ghi dữ liệu nhị phân.....	147
5.4.	File truy cập ngẫu nhiên (Random Access Files)	150
5.5.	Sử dụng luồng ký tự.....	152
5.5.1.	Nhập Console dùng luồng ký tự.....	153
5.5.2.	Xuất Console dùng luồng ký tự.....	155

5.5.3.	Đọc/ghi File dùng luồng ký tự.....	156
5.6.	Lớp File.....	158
Chương 6. ỨNG DỤNG GIAO TIẾP VỚI CƠ SỞ DỮ LIỆU.....		163
6.1.	Giới thiệu	163
6.2.	Kiến trúc JDBC, cách tạo ứng dụng JDBC.....	163
6.3.	Các khái niệm cơ bản.....	163
6.3.1.	JDBC Driver	164
6.3.2.	JDBC URL	166
6.4.	Kết nối cơ sở dữ liệu với JDBC	166
6.4.1.	Đăng ký trình điều khiển	166
6.4.2.	Thực hiện kết nối.....	167
6.4.3.	Các ví dụ.....	168
6.5.	Cách tạo truy vấn và các kiểu truy vấn.....	172
6.6.	Mô tả Rowset, JDBCRowset và CaughtRowset	178
CHƯƠNG 7. LẬP TRÌNH ỨNG DỤNG MẠNG VỚI SOCKET.....		183
7.1.	Giới thiệu chung.....	183
7.2.	Lập trình thao tác với địa chỉ máy trạm.....	183
7.2.1.	Lập trình thao tác với địa chỉ IP, lớp URL Connection	183
7.2.2.	Ví dụ sử dụng các phương thức lớp InetAddress	189
7.3.	Lập trình ứng dụng mạng với TCP socket.....	190
7.3.1.	Lớp Socket và lớp Server Socket	190
7.3.2.	Kỹ thuật lập trình truyền thông với giao thức TCP	194
7.3.3.	Một số ví dụ.....	197
7.4.	Lập trình ứng dụng mạng với UDP Socket.....	200
7.4.1.	Một số lớp Java hỗ trợ lập trình với UDP Socket	200
7.4.2.	Kỹ thuật lập trình truyền thông với giao thức UDP	205
7.5.	Lấy dữ liệu web.....	218
7.5.1.	URL và URI.....	218
7.5.2.	Sử dụng URLConnection để truy xuất dữ liệu.....	218
7.6.	Gửi Email	218
7.6.2.	Khái niệm Session	220
7.6.3.	Cách tạo message.....	220
7.6.4.	Các bước để tạo và gửi message.....	220
7.6.5.	Các bước cần thiết để đọc một message.....	221

7.6.6. Các bước cần thiết để trả lời một email.....	222
TÀI LIỆU THAM KHẢO	226

TaiLieu.vn

DANH MỤC THUẬT NGỮ TIẾNG ANH

Từ	Nghĩa của từ
Abstract	Trừu tượng
Assert	Được sử dụng để xác định vị trí lỗi chương trình nội bộ
Break	Dừng vòng lặp
Catch	Từ khóa đầu của một khối bắt ngoại lệ
Continue	Bỏ qua phần cuối vòng lặp, tiếp tục sang bước tiếp theo.
Default	Giá trị mặc định của hàm switch()
Extends	Kế thừa
Final	Một hằng số, phương thức hay một lớp không được ghi đè
Finally	Một phần của khối xử lý ngoại lệ try luôn được thực hiện
Implements	Định nghĩa giao diện mà lớp thực hiện
Import	Khai báo một gói thư viện
Instanceof	Kiểm tra một đối tượng là một thể hiện của lớp
Interface	Giao diện
Native	Một phương thức được thực hiện bởi hệ thống máy chủ
new	Tạo một đối tượng mới của lớp
Null	Tham chiếu rỗng
Package	Gói
Private	Tiền tố chỉ được truy cập bởi phương thức của lớp
Protected	Tiền tố được truy cập bởi phương thức của lớp, lớp con của và các lớp khác trong cùng một gói
Public	Tiền tố có thể được truy cập bởi phương thức của tất cả các lớp
Return	Trả về của một phương thức
Super	Gọi phương thức khởi tạo của lớp cha
Synchronized	Cho biết đây là một phương thức đồng bộ
This	Sử dụng như một tham chiếu đến đối tượng hiện tại

LỜI NÓI ĐẦU

Cùng với sự phát triển của khoa học, kỹ thuật, công nghệ thông tin ở nước ta trong những năm gần đây phát triển mạnh, đặc biệt là ngành công nghệ phần mềm. Sau khi ra đời, ngôn ngữ lập trình Java được sử dụng rộng rãi và phổ biến đối với các lập trình viên chuyên nghiệp cũng như các nhà phát triển phần mềm. Công nghệ Java đã được đưa vào giảng dạy sinh viên công nghệ thông tin ở một số trường Đại Học và các cơ sở đào tạo lập trình viên chuyên nghiệp.

Để đáp ứng với yêu cầu học tập của sinh viên chuyên ngành công nghệ thông tin, chúng tôi biên soạn giáo trình “Công nghệ Java”. Đây là học phần cơ sở của sinh viên chuyên ngành Đại học và Cao đẳng Công nghệ Thông tin. Học phần cung cấp cho sinh viên những kiến thức cơ bản và chuyên sâu về ngôn ngữ Java, trang bị cho sinh viên những kỹ năng viết phần mềm ứng dụng và phần mềm nhúng. Đây có thể xem là những kiến thức nền tảng cho các lập trình viên về công nghệ Java.

Nội dung của giáo trình bao gồm 7 chương:

Chương 1. Giới thiệu tổng quan về ngôn ngữ lập trình Java. Cách dịch và thực thi một chương trình, kiến trúc chương trình xây dựng trên Java và giới thiệu về lập trình hướng đối tượng.

Chương 2. Trình bày những khái niệm căn bản về lập trình Java như các kiểu dữ liệu và toán tử cơ bản trong Java, các câu lệnh và cấu trúc lệnh trong Java.

Chương 3. Trình bày chi tiết các khái niệm về lớp, lớp nội, đối tượng, kế thừa... tính đa hình, xử lý ngoại lệ, xử lý đa luồng và tương tranh.

Chương 4. Giới thiệu lập trình giao diện bằng SWING như: các thành phần, cách trình bày giao diện, xử lý các sự kiện...

Chương 5. Trình bày về luồng và tập tin như luồng byte, luồng ký tự, ..

Chương 6. Trình bày thiết kế ứng dụng liên quan đến cơ sở dữ liệu: như kiến trúc JDBC, cách kết nối cơ sở dữ liệu với JDBC và các thao tác trên cơ sở dữ liệu. Kết thúc chương này sinh viên có thể viết phần mềm hoàn thiện giải quyết các bài toán quản lý.

Chương 7. Giới thiệu lập trình mạng bằng socket, sử dụng các lớp được cung cấp trong JAVA. Đây là kiến thức nền hỗ trợ cho học phần Thực tập lập trình mạng học ở học kỳ tiếp theo.

Trong mỗi chương đều có các câu hỏi ôn tập và câu hỏi thảo luận để tổng kết và kiểm tra lại các kiến thức trong mỗi chương.

Chúng tôi hy vọng nội dung giáo trình sẽ giúp cho sinh viên những kiến thức cần thiết, làm cơ sở để có thể đi sâu vào thiết kế các phần mềm. Giáo trình được dùng để giảng dạy cho sinh viên ngành Công nghệ thông tin, song cũng có thể giúp sinh viên các ngành điện tử, viễn thông của các trường đại học cần tham khảo các vấn đề về lập trình Java.

Chúng tôi xin chân thành cảm ơn Thầy Nguyễn Hoàng Chiến, phó chủ nhiệm, phụ trách khoa Công nghệ Thông tin trường Đại học Kinh tế Kỹ thuật Công nghiệp cùng với các đồng nghiệp đã đóng góp nhiều ý kiến quý báu cho sự thành công của cuốn tài liệu này. Vì tài liệu được biên soạn lần đầu, chúng tôi đã rất cố gắng hoàn chỉnh, song không tránh khỏi thiếu sót. Rất mong nhận được sự góp ý của bạn đọc để tài liệu học tập được hoàn thiện hơn.

Xin trân trọng cảm ơn!

Tác giả

Vũ Văn Đốc

Lương Thị Thảo Hiếu

Lê Thanh Cửa

Chương 1. TỔNG QUAN VỀ LẬP TRÌNH JAVA

Mục đích

Nội dung tập trung trình bày các vấn đề chính về ngôn ngữ lập trình Java:

- Lịch sử ra đời và phát triển của Java
- Kiến trúc tổng quát một chương trình xây dựng trên Java
- Các đặc điểm của Java, khái niệm máy ảo.
- Cấu trúc một chương trình Java đơn giản, cách xây dựng, dịch, thực thi một chương trình Java.

1.1. Lịch sử ra đời và phát triển của Java

Năm 1991, một nhóm kỹ sư của Sun Microsystems muốn lập trình điều khiển các thiết bị điện tử như tivi, máy giặt... Ban đầu, định dùng C và C++ nhưng trình biên dịch C/C++ phụ thuộc vào từng loại CPU. Do đó, họ đã bắt tay vào xây dựng một ngôn ngữ chạy nhanh, gọn, hiệu quả, độc lập thiết bị và ngôn ngữ “Oak” ra đời vào năm 1995, tương tự như C++ nhưng loại bỏ một số tính năng nguy hiểm của C++ và có khả năng chạy trên nhiều nền phần cứng khác nhau. Cùng lúc đó world wide web bắt đầu phát triển và Sun thấy được tiềm năng của ngôn ngữ Oak nên đã đầu tư cải tiến và phát triển.

Ngôn ngữ lập trình Java được Sun Microsystems giới thiệu vào tháng 6 năm 1995 và nhanh chóng trở thành một ngôn ngữ lập trình của các lập trình viên chuyên nghiệp. Java được xây dựng dựa trên nền tảng của C và C++, Java sử dụng cú pháp của C và đặc trưng hướng đối tượng của C++. Java là ngôn ngữ vừa biên dịch vừa thông dịch. Đầu tiên mã nguồn được biên dịch thành dạng bytecode. Sau đó được thực thi trên từng loại máy nhờ 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 các nền phần cứng khác nhau.

Ngày nay, Java được sử dụng rộng rãi, không chỉ để viết ứng dụng trên máy cục bộ hay trên mạng mà còn để xây dựng các trình điều khiển thiết bị di động, PDA,...

1.2. Đặc trưng ngôn ngữ Java

Tính đơn giản

Ngôn ngữ lập trình Java loại bỏ các đặc trưng phức tạp của C và C++ như: Loại bỏ thao tác con trỏ, thao tác định nghĩa chồng toán tử (operator overloading)..., không cho phép đa kế thừa (Multi-inheritance) mà sử dụng giao diện (interface), không sử dụng lệnh “goto” cũng như file header (.h), loại bỏ cấu trúc “struct” và “union”.

Tính hướng đối tượng

Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương trình viết

bằng Java (hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ thay vào đó Java đưa ra khái niệm interface hỗ trợ tính đa kế thừa.

Tính độc lập với phần cứng và hệ điều hành

Tính độc lập với phần cứng được hiểu theo nghĩa một chương trình Java, nếu chạy đúng trên phần cứng của một họ máy nào đó thì cũng chạy đúng trên tất cả các họ máy khác. Một chương trình chỉ chạy đúng trên một số họ máy cụ thể được gọi là phụ thuộc vào phần cứng.

Tính độc lập với hệ điều hành được hiểu theo nghĩa một chương trình Java có thể chạy được trên tất cả các hệ điều hành. Một chương trình chỉ chạy được trên một số hệ điều hành được gọi là phụ thuộc hệ điều hành.

Các chương trình viết bằng Java có thể chạy trên hầu hết các hệ nền mà không cần phải thay đổi gì, những người lập trình đặt cho nó một khẩu hiệu ‘viết một lần, chạy mọi nơi’, điều này là không thể có với các ngôn ngữ lập trình khác.

Tính mạnh mẽ

Java là ngôn ngữ yêu cầu chặt chẽ về kiểu dữ liệu, việc ép kiểu tự động bừa bãi của C, C++ được hạn chế trong Java, điều này làm chương trình rõ ràng, sáng sủa, ít lỗi hơn. Java kiểm tra lúc biên dịch và cả trong thời gian thông dịch vì vậy loại bỏ một số loại lỗi lập trình nhất định. Java không sử dụng con trỏ và các phép toán con trỏ. Java kiểm tra tất cả các truy nhập đến mảng, chuỗi khi thực thi để đảm bảo rằng các truy nhập đó không ra ngoài giới hạn kích thước.

Trong các môi trường lập trình truyền thống, lập trình viên phải tự mình cấp phát bộ nhớ. Trước khi chương trình kết thúc thì phải tự giải phóng bộ nhớ đã cấp. Trong chương trình Java, lập trình viên không phải bận tâm đến việc cấp phát. Quá trình cấp phát, giải phóng được thực hiện tự động, nhờ dịch vụ thu gom rác (garbage collection). Đồng thời cơ chế bẫy lỗi của Java giúp đơn giản hóa quá trình xử lý lỗi và hồi phục sau lỗi.

Tính an toàn

Java cung cấp một môi trường quản lý thực thi chương trình với nhiều mức để kiểm soát tính an toàn:

Ở mức thứ nhất, mức ngôn ngữ: Dữ liệu và các phương thức được đóng gói bên trong lớp. Chúng chỉ được truy xuất thông qua các giao diện mà lớp cung cấp.

Ở mức thứ hai, mức Compiler: Trình biên dịch kiểm tra an toàn cho code trước khi biên dịch.

Mức thứ ba được đảm bảo bởi trình thông dịch: Chúng kiểm tra xem bytecode có đảm bảo các qui tắc an toàn trước khi thực thi.

Mức thứ tư, mức class: 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.

Tính phân tán

Java được thiết kế để hỗ trợ các ứng dụng chạy trên mạng bằng các lớp mạng nằm trong gói (Java.net). Cụ thể là Java có hỗ trợ công nghệ lập trình RMI, CORBA, JavaBean. Các công nghệ này cho phép sử dụng lại các lớp đã tạo ra, triệu gọi các phương thức (method) hoặc các đối tượng từ một máy ở xa.

Tính đa luồng

Tính năng này cho phép viết một chương trình có nhiều đoạn mã lệnh được chạy song song với nhau. Với Java có thể viết các chương trình có khả năng chạy song song một cách dễ dàng, hơn thế nữa việc đồng bộ tài nguyên dùng chung trong Java cũng rất đơn giản. Điều này không thể có đối với một số ngôn ngữ lập trình khác như C/C++, pascal...

Tính động

Java được thiết kế như một ngôn ngữ động để đáp ứng cho những môi trường mở. Các chương trình Java bổ sung các 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 ứng dụng của Java

Ứng dụng thực thi qua dòng lệnh (Console)

Ứng dụng Console là ứng dụng nhập xuất ở chế độ văn bản tương tự như màn hình Console của hệ điều hành MS-DOS. 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.

Ứng dụng Java Applet

Applet là chương trình Java được tạo ra để sử dụng trên Internet thông qua các trình duyệt hỗ trợ Java như IE hay Netscape. Applet được nhúng bên trong trang Web. Khi Web hiển thị trong trình duyệt, Applet sẽ được tải về và thực thi tại trình duyệt.

Ứng dụng giao diện (GUI Application)

Việc phát triển các chương trình ứng dụng có giao diện đồ họa trực quan 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.

Ứng dụng web

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 đang quảng cáo. Công nghệ 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.

Ứng dụng lập trình mạng

Có thể nói Java hỗ trợ rất mạnh về lập trình ứng mạng với các lớp thư viện socket giúp đơn giản hóa quá trình kết nối và chuyển dữ liệu trên mạng. Java cho phép người lập trình có thể lập trình trên rất nhiều giao thức TCP/ IP, UDP, HTTP, FTP, Telnet....

Những ứng dụng điển hình: Các chương trình chat, chương trình chuyển dữ liệu, truyền tin thông báo tin tức, trao đổi trực tuyến, gửi nhận thư điện tử....

Ứng dụng cơ sở dữ liệu

CSDL là yếu tố không thể thiếu trong hầu hết các ứng dụng, việc hỗ trợ truy xuất CSDL không phải lúc nào cũng đầy đủ trong các ngôn ngữ lập trình. Ví dụ VB cung cấp rất nhiều cơ chế truy xuất CSDL thuận tiện như ODBC, ADO, ADO.NET, OLEDB trong khi C/C++ không có.

Java hỗ trợ các cơ chế truy xuất CSDL mở với công nghệ JDBC (Java Database Connectivity), cho phép truy xuất đến nhiều loại CSDL khác nhau như Oracle, DB2, MySQL, SQL Server . JDBC cho phép truy cập dữ liệu thông qua mô hình đa luồng rất thích hợp cho các ứng dụng phân tán.

1.4. Dịch và thực thi một chương trình viết bằng Java

Việc xây dựng, dịch và thực thi một chương trình viết bằng Java có thể tóm tắt qua các bước sau:

- Viết mã nguồn: Dùng một chương trình soạn thảo nào đó (NotePad hay Jcreator chẳng hạn) để viết mã nguồn và lưu lại với tên có đuôi “. Java”.
- Biên dịch ra mã máy ảo: Dùng trình biên dịch Javac để biên dịch mã nguồn “.Java” thành mã của máy ảo (Java bytecode) có đuôi “.class” và lưu lên đĩa.
- Thông dịch và thực thi: Ứng dụng được load vào bộ nhớ, thông dịch và thực thi dùng trình thông dịch Java thông qua lệnh “Java”.
- Đưa mã Java bytecode vào bộ nhớ: đây là bước “loading”. Chương trình phải được đặt vào trong bộ nhớ trước khi thực thi. “Loader” sẽ lấy các files chứa mã Java bytecode có đuôi “.class” và nạp vào bộ nhớ.
- Kiểm tra mã Java bytecode: trước khi trình thông dịch chuyển bytecode thành mã máy tương ứng để thực thi thì mã bytecode phải được kiểm tra tính hợp lệ.

Thông dịch & thực thi: cuối cùng dưới sự điều khiển của CPU và trình thông dịch tại mỗi thời điểm sẽ có mã bytecode được chuyển sang mã máy và thực thi.

1.5. Kiến trúc chương trình xây dựng trên Java

1.5.1. Kiến trúc chương trình Java

Dạng cơ bản của một tập tin mã nguồn Java có cấu trúc như sau :

```
package packageName; // Khai báo tên gói, nếu có
import java.awt.*;    // Khai báo tên thư viện sẵn có, nếu cần dùng
class className      // Khai báo tên lớp
```

```

{
/* Đây là dòng chú thích, nếu cần */
int var;          // Khai báo biến
public void methodName() // Khai báo tên phương thức
{
    /* Phần thân của phương thức */
    // Các lệnh thực hiện trong thân phương thức
}
}

```

Một tệp mã nguồn Java có thể có ba phần chính:

- Phần khai báo tên gói (khối) bằng từ khoá package.
- Phần khai báo thư viện tham khảo bằng từ khoá import.
- Phần khai báo nội dung lớp bằng từ khoá class.

Khai báo Package

Package dùng để đóng gói các lớp trong chương trình lại với nhau thành một khối. Đây là một cách hữu hiệu để lưu trữ các lớp gần giống nhau hoặc có cùng một module thành một khối thống nhất.

Cú pháp khai báo tên gói bằng từ khoá package:

```
package <Tên gói>;
```

Để đặt tên package trong chương trình, người ta có thể tiến hành như đặt tên thư mục trên ổ đĩa. Nghĩa là bắt đầu bằng tên có phạm vi lớn, cho đến các tên có phạm vi nhỏ, cuối cùng là tên các gói trực tiếp chứa các lớp. Phạm vi đặt tên gói, trên thực tế, thường được tiến hành theo thứ tự phạm vi lớn đến nhỏ như sau:

- Tên tổ chức
- Tên công ty
- Tên dự án
- Tên modul trong dự án
- Tên các chức năng trong modul

Ưu điểm của package:

- Cho phép nhóm các lớp vào với nhau thành các đơn vị nhỏ hơn. Việc thao tác trên các đơn vị khối sẽ gọn hơn thao tác trên một tập các lớp.
- Tránh việc xung đột khi đặt tên lớp. Khi số lượng lớp của chương trình quá lớn ta có thể tránh phải đặt tên khác nhau cho các lớp bằng cách đặt chúng vào các package khác nhau.

- Cho phép bảo vệ các lớp. Khi chương trình lớn, việc chia nhỏ chương trình thành các package sẽ thuận lợi hơn cho việc quản lý và phát triển.
- Tên gói còn được dùng để định danh lớp trong ứng dụng.

Lưu ý:

- Các tệp tin của các lớp nằm cùng gói ứng dụng phải được lưu trong cùng một thư mục (tên thư mục là tên khối) theo cấu trúc khối của dự án.
- Tên gói nên đặt theo chữ thường vì tên gói sẽ là tên thư mục tương ứng trong ổ đĩa, tránh nhầm lẫn với tên các tệp tin là tên các lớp của chương trình.

Khai báo thư viện

Khai báo thư viện để chỉ ra những thư viện đã được định nghĩa sẵn mà chương trình sẽ tham khảo tới. Cú pháp khai báo thư viện với từ khoá import như sau:

import <Tên thư viện>;

Java chuẩn cung cấp một số thư viện như sau:

- *java.lang*: cung cấp các hàm thao tác trên các kiểu dữ liệu cơ bản, xử lý lỗi và ngoại lệ, xử lý vào ra trên các thiết bị chuẩn như bàn phím và màn hình.
- *java.applet*: cung cấp các hàm cho xây dựng các applet
- *java.awt*: cung cấp các hàm cho xây dựng các ứng dụng đồ hoạ với các thành phần giao diện đa phương tiện multimedia.
- *java.io*: cung cấp các hàm xử lý vào/ra trên các thiết bị chuẩn, các thiết bị ngoại vi.
- *java.util*: cung cấp các hàm tiện ích trong xử lý liên quan đến các kiểu dữ liệu có cấu trúc như Date, Stack, Vector.

Lưu ý:

- Nếu muốn khai báo tham khảo nhiều thư viện, phải khai báo tham khảo mỗi thư viện với một từ khoá import.
- Nếu chỉ tham khảo một vài lớp trong một thư viện, nên chỉ rõ tham khảo lớp nào, thay vì phải khai báo tham khảo cả gói (bằng kí hiệu “*”) vì tham khảo cả gói sẽ tăng kích cỡ tệp tin class sau khi biên dịch.
- Nếu không tham khảo thư viện nào, không cần thiết phải khai báo các tham khảo với từ khoá import.

Khai báo lớp

Phần thứ ba là phần khai báo lớp và nội dung của lớp, phần này luôn bắt buộc phải có đối với một tệp mã nguồn Java:

- Khai báo tên lớp với từ khoá class.
- Khai báo các thuộc tính của lớp.
- Khai báo các phương thức của lớp

1.5.2. Chương trình Java đầu tiên.

1.5.2.1. Công cụ soạn thảo mã nguồn Java.

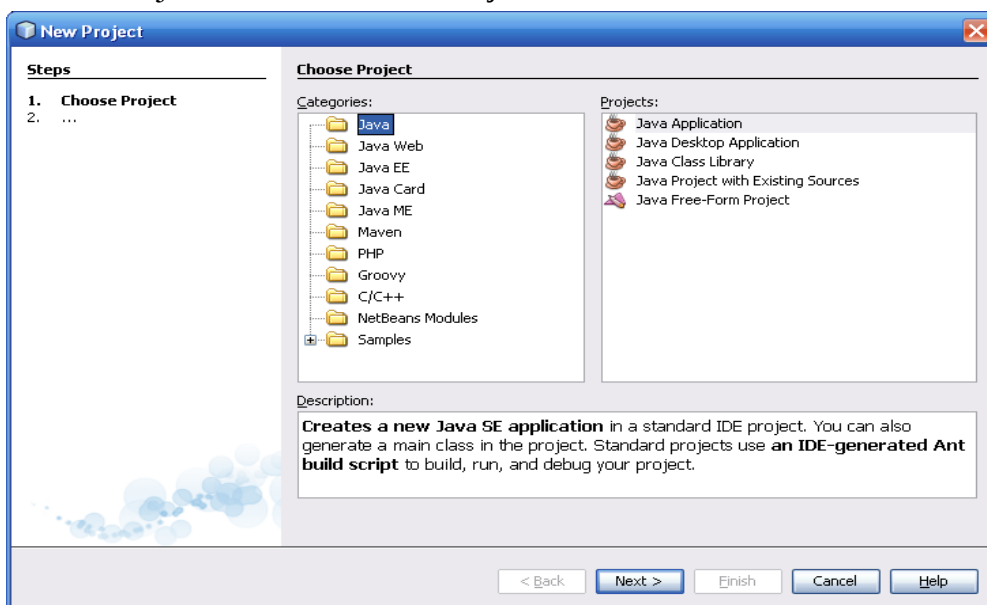
Để viết mã nguồn Java chúng ta có thể sử dụng trình soạn thảo NotePad hoặc một số môi trường phát triển hỗ trợ ngôn ngữ Java như: Jbuilder của hãng Borland, JDeveloper của hãng Oracle, Visual J++ của Microsoft...

Trong khuôn khổ giáo trình này để hướng dẫn sinh viên thực hành chúng tôi dùng công cụ NetBeans IDE 7.0 được phát triển bởi Sun Microsystems.

NetBean 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. Chương trình được viết bằng Java nhưng có thể hỗ trợ bất kỳ ngôn ngữ lập trình nào. Với giao diện thân thiện, được coi là phổ biến và hiệu quả nhất, một sản phẩm miễn phí và không có giới hạn trong việc sử dụng, NetBean IDE là lựa chọn tối ưu nhất để viết Java. NetBean IDE dễ dàng cài đặt và chạy trên nhiều hệ điều hành, bao gồm Windows, Linux, hệ điều hành Mac OS X và Solaris... có thể download Netbean IDE tại: netbeans.org/downloads/

Ví dụ. Để tạo và thực thi chương trình có tên HelloWorldApp như sau:

Bước 1. Tạo Project: Chọn File/NewProject:

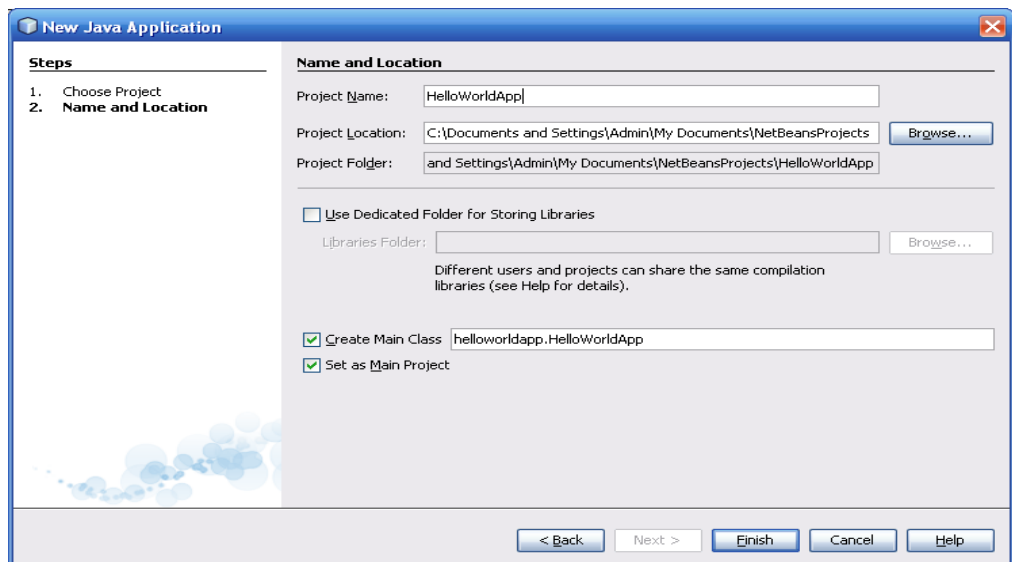


Hình 1.1. Tạo project mới

Tại mục Categories ta chọn Java.

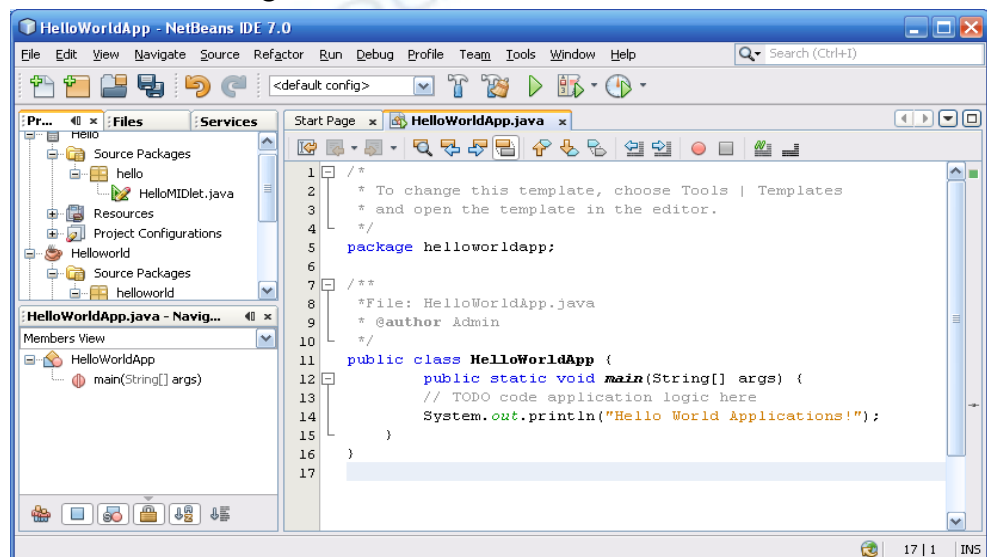
Tại mục Projects ta chọn Java Application rồi chọn Next.

Nhập Project Name và chọn Finish.



Hình 1. 2. Vị trí lưu project

Bước 2. Soạn thảo mã nguồn tại cửa sổ bên dưới:



Hình 1.3. Màn hình soạn thảo

1.5.2.2. Chương trình Java.

Chương trình sau cho phép hiển thị thông điệp: "Hello World Application!"

```
package helloworldapp;

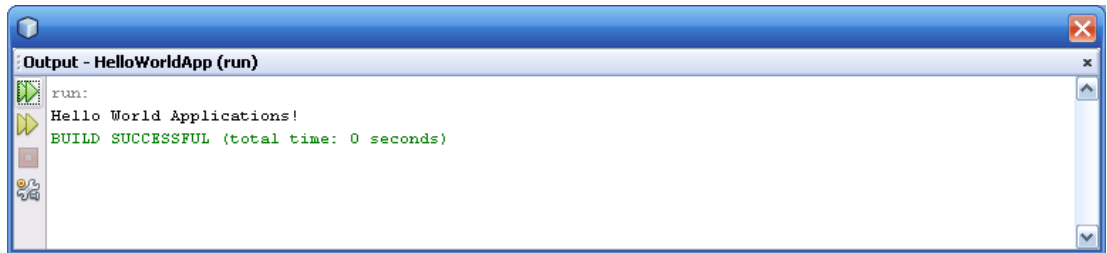
/**
 *File: HelloWorldApp.Java
 * @author Admin
 */
public class HelloWorldApp {
    /**
     * @param args the command line arguments
     */
}
```

```

public static void main(String[] args) {
    // TODO code application logic here
    System.out.println("Hello World Applications!");
}
}

```

Để thực thi chương trình ta vào Run/Run Main Project hoặc Bấm Run Main Project () trên thanh công cụ hoặc bấm phím F6. Kết quả hiển thị trên màn hình:



1.5.2.3. Phân tích chương trình đầu tiên.

Tập tin HelloWorldApp.Java sẽ nằm trong gói helloworldapp

```

/*
 *File: HelloWorldApp.Java
 * @author Admin*/

```

Ký hiệu “/*... */” dùng để chú thích nhiều dòng lệnh. Trình biên dịch sẽ bỏ qua các dòng chú thích này. Java hỗ trợ hai loại chú thích:

- Loại chú thích trên một dòng, dùng “//”. Trình biên dịch sẽ bỏ qua nội dung bắt đầu từ ký hiệu “//” cho đến hết dòng lệnh chứa nó.
- Loại chú thích trên nhiều dòng có thể bắt đầu với “/*” và kết thúc với “*/”. Trình biên dịch sẽ bỏ qua nội dung nằm giữa hai ký hiệu này.

Dòng kế tiếp khai báo lớp có tên HelloWorldApp: Bắt đầu với từ khoá public, kế đến là từ khóa class, cuối cùng là tên lớp:

```
public class HelloWorldApp {...}
```

Một định nghĩa lớp nằm trọn vẹn giữa hai ngoặc móc mở “{” và đóng “}”. Các ngoặc này đánh dấu bắt đầu và kết thúc một khối lệnh.

```
public static void main(String args[ ])
```

Đây là phương thức chính, từ đây chương trình bắt đầu việc thực thi của mình. Tất cả các ứng dụng Java đều sử dụng một phương thức main này.

- Từ khoá public là một chỉ định truy xuất. Nó cho biết thành viên của lớp có thể được truy xuất từ bất cứ đâu trong chương trình.
- Từ khoá static cho phép main được gọi tới mà không cần tạo ra một thể hiện (instance) của lớp. Nó không phụ thuộc vào các thể hiện của lớp được tạo ra.

- Từ khoá void thông báo cho máy tính biết rằng phương thức sẽ không trả lại bất cứ giá trị nào khi thực thi chương trình.
- String args[] là tham số dùng trong phương thức main. Khi không có một thông tin nào được chuyển vào main, phương thức được thực hiện với các dữ liệu rỗng - không có gì trong dấu ngoặc đơn.
- System.out.println("Hello World Applications!"); Dòng lệnh này hiển thị chuỗi "Hello World Applications!" trên màn hình.
- Lệnh println() cho phép hiển thị chuỗi được truyền vào lên màn hình.

TaiLieu.vn

Chương 2. CẤU TRÚC LẬP TRÌNH CƠ BẢN TRONG JAVA

Mục đích

Nội dung chương này trình bày khái niệm cơ bản về lập trình trong Java như:

- Các kiểu dữ liệu, cách khai báo biến và các toán tử.
- Cách nhập xuất dữ liệu
- Các cấu trúc lệnh trong Java như các cấu trúc lựa chọn và các vòng lặp
- Mảng và các thao tác cơ bản trên kiểu dữ liệu mảng trong Java

2.1. Các kiểu dữ liệu

Kiểu dữ liệu cơ bản

- Kiểu số nguyên.

Kiểu dữ liệu	Kích	Phạm vi biểu diễn	Giá trị
byte	8 bits	$-128 \div 127$	0
short	16 bits	$-32768 \div 32767$	0
int	32 bits	$-2^{31} \div 2^{31}-1$	0
long	64 bits	$-2^{63} \div 2^{63}-1$	0l

- Kiểu số thực

Kiểu dữ liệu	Kích	Phạm vi biểu diễn	Giá trị
float	32 bits	$-3.4e38 \div 3.4e38$	0.0f
double	64 bits	$-1.7976...E+308 \div +1.7976...E+308$	0.00d

- Kiểu logic (boolean)

Nhận giá trị true hoặc false. Giá trị mặc định là false

Kiểu ký tự (char)

Kiểu dữ liệu	Kích	Phạm vi biểu diễn	Giá trị
char	16 bits	$\backslash u0000 \div \backslash uffff$	0

Kiểu dữ liệu đối tượng :

Trong Java, có 3 kiểu dữ liệu đối tượng:

- **array:** Một mảng của các dữ liệu cùng kiểu
- **class:** Dữ liệu kiểu lớp đối tượng do người dùng định nghĩa.
- **interface:** Dữ liệu kiểu lớp giao diện do người dùng định nghĩa. Chứa các phương thức của giao diện.

Ép kiểu Chuyển đổi giữa các kiểu dữ liệu cơ sở: