

Chapter 1 – Introduction to Computers, the Internet, and the Web

Java technologies are classified into three editions:

1. Standard (J2SE technology)
2. Micro (J2ME technology) Advanced Java 2 Platform How to program
3. Enterprise (J2EE technology) 多了EJB

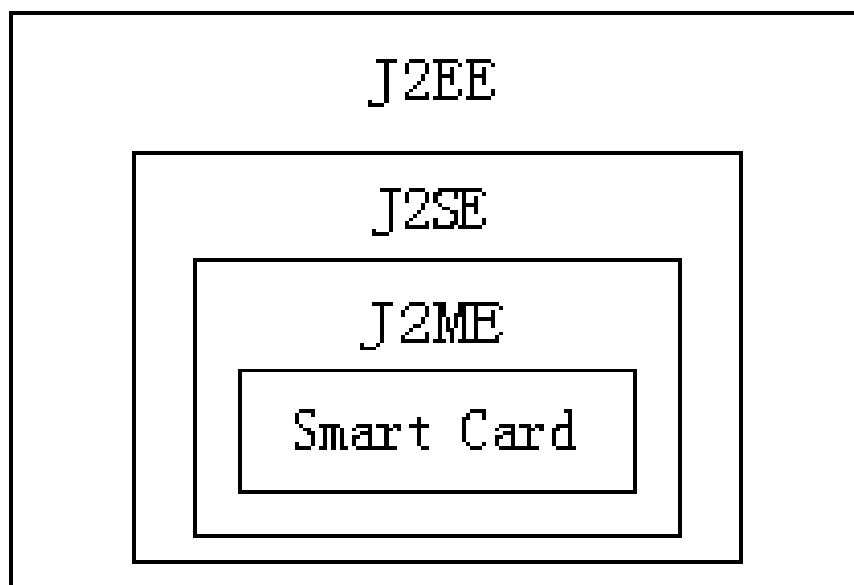
Each edition is a developer treasure chest of tools and supplies that can be used with a particular product:

1. Java virtual machines
2. A library of APIs
3. Tools for deployment and device configuration
4. A profile that is a specification of the set of APIs

Reference: <http://java.sun.com>

1

Java 主要版本範圍圖



2

Evolution of Operating Systems

- **Single user Batch processing**
 - One job (task) at a time for CPU
 - Operating systems developed
 - Programs to make computers more convenient to use
 - Switch jobs easier(one after another)
 - Increase throughput
 - Monopoly computer resources
- **Multiprogramming**
 - Share computer resources
 - Increase utilization
 - “Simultaneous” jobs
 - Timesharing operating systems
 - CPU time slice
 - User receives immediate response

3

Personal, Distributed and Client/Server Computing

- **Personal computing**
 - Computers for personal use
- **Distributed computing**
 - Computing performed among several computers
- **Client/server computing**
 - Servers offer common store of programs and data
 - Clients access programs and data from server
- **3 Tiers architecture**
 - Client/Server(pure Homepage) + Server DB(provide data)

4

Machine Languages, Assembly Languages and High-Level Languages

- Machine language
 - “Natural language” of computer component
 - Machine dependent
- Assembly language
 - English-like abbreviations represent computer operations
 - Translator programs convert to machine language
- High-level language
 - Allows for writing more “English-like” instructions
 - Contains commonly used mathematical operations
 - Compiler convert to machine language
- Interpreter
 - Execute high-level language programs without compilation

5

History of C++

- C++
 - Evolved from C
 - Evolved from BCPL and B
 - Provides object-oriented programming capabilities
- Objects
 - Reusable software components that model real-world items

6

History of Java

- Java
 - Originally for intelligent consumer-electronic devices
 - Then used for creating Web pages with *dynamic content*
 - Now also used for:
 - Develop large-scale enterprise applications
 - Enhance WWW server functionality
 - Provide applications for consumer devices (cell phones, etc.)

7

Java Class Libraries

- Classes
 - Contain *methods* that perform tasks
 - Return information after task completion
 - Used to build Java programs
- Java contains class libraries
 - Known as Java APIs (Application Programming Interfaces)

8

FORTRAN, COBOL, Pascal and Ada

- Fortran
 - FORMula TRANslator
- COBOL
 - COMmon Business Oriented Language

9

Structured Programming

- Structured Programming
 - Structured programs
 - Clearer than unstructured programs
 - Easier to test, debug and modify
 - Pascal designed for teaching structured programming
 - ADA
 - Multitasking(**rendezvous**) → allow programmers to specify many activities are occurred in parallel
 - C
 - Java
 - Multithreading → enables programmers to write programs with parallel activities

10

BASIC, Visual Basic, Visual C++, C# and .NET

- BASIC
 - Beginner's All-Purpose Symbolic Instruction Code
- Visual Basic .NET
 - Framework Class Library (FLC)
- Visual C++
 - Microsoft Foundation Classes (MFC)
- C#
 - C-Sharp
- .NET
 - .NET Framework

11

The Internet and the World Wide Web

- Internet
 - Developed more than four decades ago with DOD funding
 - Originally for connecting few main computer systems
 - Now accessible by hundreds of millions of computers
- World Wide Web (WWW)
 - Allows for locating/viewing multimedia-based documents

12

Basics of a Typical Java Environment

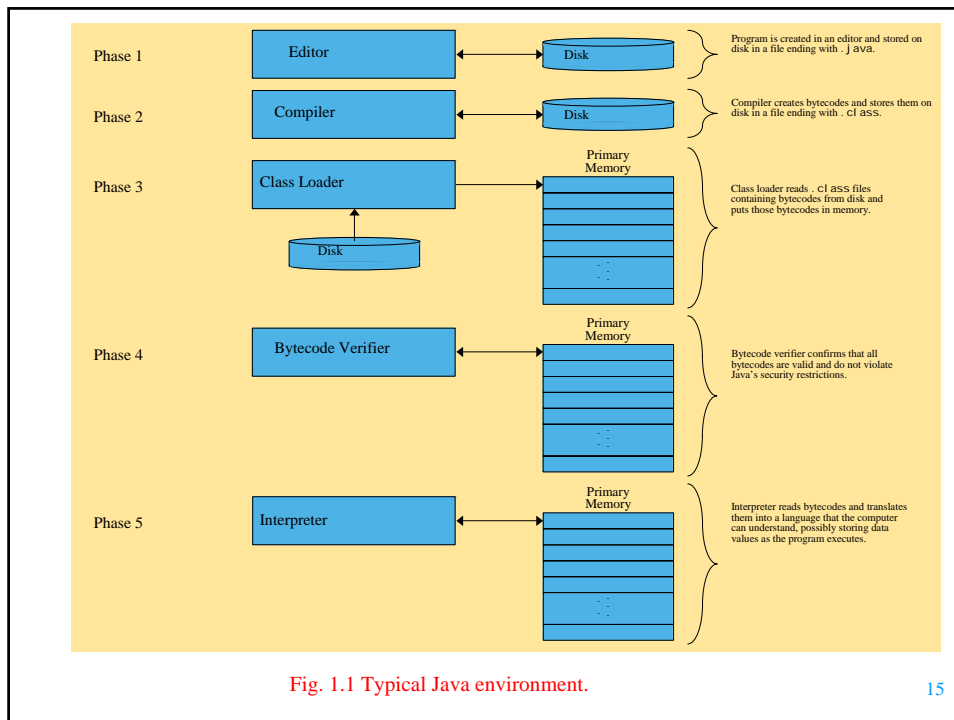
- Java systems contain
 - Environment
 - Language
 - APIs
 - Class libraries

13

Basics of a Typical Java Environment (cont.)

- Java programs normally undergo five phases
 - Edit (*.java)
 - Programmer writes program (and stores program on disk)
 - Compile (*.class) → javac [machine independent]
 - Compiler creates *bytecodes* from program
 - Load (**J**ust-**I**n-**T**ime)
 - Class loader stores bytecodes in memory
 - Verify
 - Verifier ensures bytecodes do not violate security requirements
 - Execute (Java Run-Time Environment) → java [machine dependent]
 - Interpreter translates bytecodes into machine language
- J2SDK <http://java.sun.com/Download5>
- Forte <http://www.sun.com/forte/ffj>

14



Freeware & Shareware

- **Open source**
 - **Freeware**
 - Download from internet and use as free SW
 - Subject to the restrictions specified by the copyright owner
 - **Shareware**
 - Download from internet and use as free SW for personal use
 - Pay a fee to the copyright owner for commercial products
 - **Open source community**
 - Download the source code from internet and learn from them
 - Make some enhancements and publish to the open source community
 - ➔ Linux is an example of open source community

Java programming environment

- Java IDEs(Integrated Development Environments)
 - Borland's JBuilder
 - Symantec's Visual Café
 - IBM's VisualAge (WebSphere)
 - Java Creator (<http://www.jcreator.com>)
 - NetBean (<http://www.netbeans.org/>)
 - Eclipse (<http://eclipse.cis.sinica.edu.tw/downloads/index.php>)

17

Java program developing categories

- Application → can execute and save standalone
 - A program is stored and executed from the user's local computer
 - Execute result can be display in DOS or open a window dialog(Swing)
- Applet(client load heavy) → embedded in HTML
 - A small program is stored on a remote computer
 - Users connect to it from WWW browser
 - Applets are downloaded from server computer to the client browser and are execute and display result in client's browser
 - After executing the Applet, client's browser discards the applet(byte code), reload again if the client's browser wants to execute again
- Servlet (Server load heavy) – troublesome
- JSP
 - ASP like

18

Servlets與JSP

- Servlets是首先發展出來的動態網頁技術，功能非常強大而且完善，但是在產生頁面時，必需使用out.println()一句一句的輸出
- 在網頁輸出的程式片段，Servlets必需靠out.println()來輸出HTML標籤，如果是內容複雜的網頁，不難想像製作與維護的困難。
- 同樣是在網頁顯示"Hello World!"，使用JSP技術來製作的話，使用HTML標籤配合JSP程式碼，可以大幅度簡化程式

19

JSP與ASP的比較

	ASP	JSP
網頁伺服器	IIS or PWS	Apache, and IIS
平台	Microsoft Windows	大部份常見的平台, 包含 Solaris Operating Environment, Microsoft Windows, Mac OS, Linux, 和其它 UNIX platform implementations
可重複使用, 跨平台的元件	No	JavaBeans, custom JSP tags
安全性與防止系統毀損	No	Yes
記憶體漏失保護	No	Yes
使用的描述語言	VBScript, JScript	Java
自定標籤	No	Yes

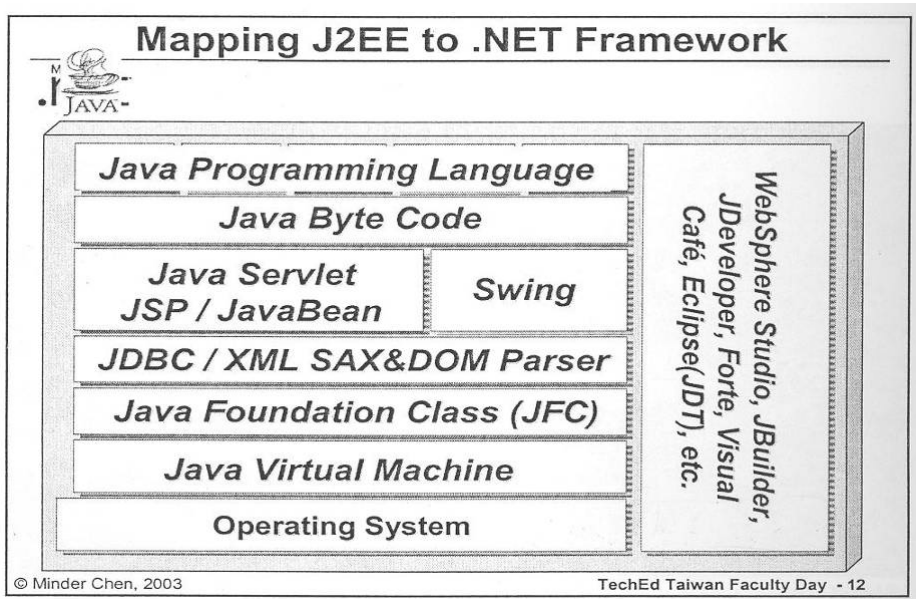
20

General Notes about Java and This Book

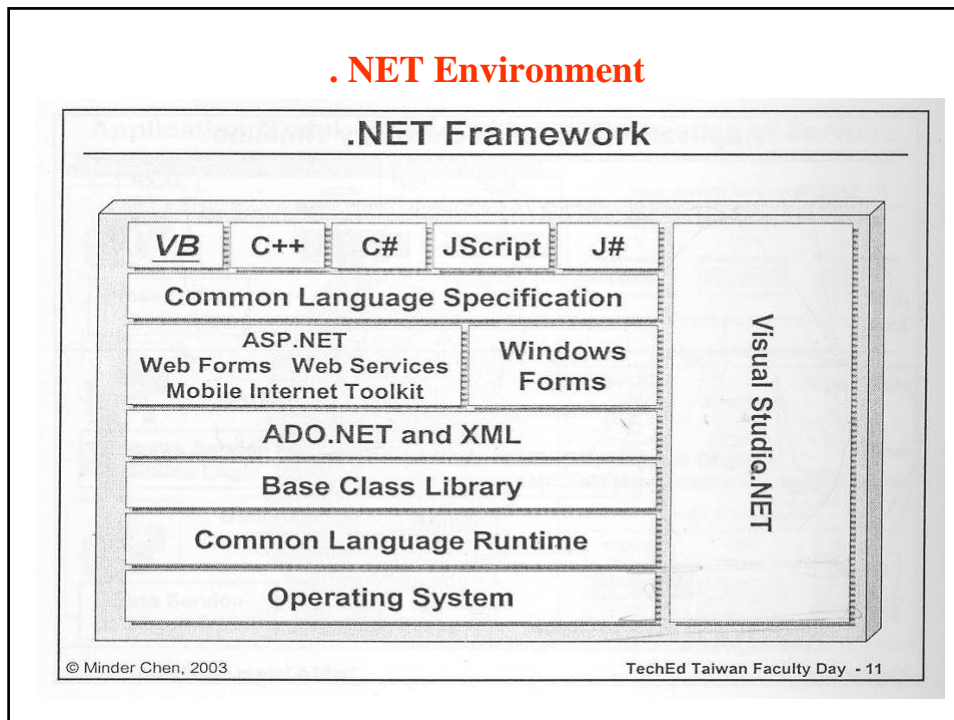
- Geared for novice programmers
- We stress clarity
 - Comments appear in **Green**
 - Keywords appear in **Blue**
 - Constants appear in **Light Blue**
 - Class and variable names appear in **Black**

21

Java Environment



.NET Environment



Thinking About Objects: Introduction to Object Technology and the Unified Modeling Language

- Object orientation(has attributes and exhibit behaviors)
- Unified Modeling Language (UML)
 - Graphical language that uses common notation
 - Allows developers to represent object-oriented designs
- Objects
 - Reusable software components that model real-world items
 - Look all around you
 - People, animals, plants, cars, etc.
 - Characteristics(attributes)
 - Size, shape, color, weight, etc.
 - Behaviors(methods)
 - Babies cry, crawl, sleep, etc.

Thinking About Objects (cont.)

- Object-oriented design (OOD)
 - Models real-world objects
 - Models communication among objects(via messages)
 - *Encapsulates* data (attributes) and functions/methods (behaviors)
 - Information hiding
 - Communication through well-defined interfaces
- Object-oriented language
 - Programming is called *object-oriented programming (OOP)*
 - Java
- Object-Oriented Analysis and Design (OOAD)
 - Essential for large programs
 - Analyze program requirements, then develop solution
 - We begin OOAD in Chapter 2
 - Elevator-simulation case study

25

Thinking About Objects (cont.)

- History of the UML
 - Need developed for process with which to approach OOAD
 - Union of Booch, Rumbaugh and Jacobson
 - Object Management Group (OMG) supervised in 1997
 - Version 1.4 is current version in 2001 (<http://www.omg.org/uml>)
 - Version 2.0 scheduled tentatively for release in 2002
- UML
 - Graphical representation scheme(**use-case, class, object, sequence, collaborative, statechart, activity, component, deployment diagrams**)
 - Enables developers to model object-oriented systems
 - Flexible and extendible

26

Discovering Design Patterns: Introduction

- Effective design crucial for large programs
- Design patterns
 - Proven architectures for developing object-oriented software
 - Architectures created from accumulated industry experience
 - Reduce design-process complexity
 - Promotes design reuse in future systems
 - Helps identify common design mistakes and pitfalls
 - Helps design independently of implementation language
 - Establishes common design “vocabulary”
 - Shortens design phase in software-development process
 - Similar to architectural elements
 - arches and columns
 - Used by developers to construct sets of classes and objects
- Developers
 - Familiarity with patterns to understand how to use patterns

27

Discovering Design Patterns (cont.)

- History of Design Patterns
 - Gamma, Helm, Johnson and Vlissides
 - “Gang of Four”
 - *Design Patterns, Elements of Reusable Object-Oriented Software* (Addison Wesley: 1995)
 - Established 23 design patterns
 - Creational
 - Instantiate objects
 - Structural
 - Organize classes and objects
 - Behavioral
 - Assign responsibilities to objects

28