JAVA Overview

Vittayasak Rujivorakul

Lecture 1

JAVA Overview

Vittayasak Rujivorakul

Lecture 1

Course outline

- JAVA Overview
- Introduction to Object
- Program Flow Controling
- Initialization and Cleanup
- Hiding the Implementation
- Reusing Classes
- Polymorphism Your Object

Couse Outline

- Error Handling With Exceptions
- Threads
- I/O System and Files
- Windows (GUI) & Applet
- Java Foundation Class
- JDBC
- Network Programming

1991 Software microcontrol

- **ในยุคแรกของการเขียน software micro control ใช้ C,C++ แต่พบว่ามีข้อเสียคือ**
  - การใช้ pointer ทำให้เกิดข้อผิดพลาดได้ง่าย
  - ใช้ memory จำนวนมาก
  - ไม่มีการทำ type checking
  - อิงตาม platform processor ที่สร้างโปรแกรม
- **ต่อมาได้มีการพัฒนาภาษาใหม่ขึ้นมาชื่อว่า Oak**
  - มีการทำงานเป็น emulator interpreter
  - platform independent

1992 Set-top box

- **บริษัท FirstPerson Inc. ลูกข่ายของ Sun ได้สร้าง Set-top box ขึ้นมาโดยใช้งานกับ**
  - Video On demand
  - Credit card
  - Machine control
  - CDROM
- **แต่ไม่ได้รับการสนับสนุนจากบริษัทผู้ผลิต จึงต้องพักโครงการนี้ไว้** (ในช่วงนั้น Sun ยังไม่ได้มองเรื่องการนำมาใช้งานกับ internet)

1993 HTML เกิดขึ้นมา

- **เริ่มมีการนำ HTML มาใช้งานอย่างแพร่หลาย**
- **มีปัญหาในการสร้าง application ที่สามารถเปิดข้าม platform ได้**
- **ทาง Sun จึงได้นำ Oak มาพัฒนาจนเป็น JAVA ในปี 1995**

1995 Hot Java

- **ถือว่า Java ได้กำเนิดอย่างเต็มตัวแล้วโดยมีคุณลักษณะหลักต่อไปนี้**
  - มีพื้นฐานมาจาก C,C++ ทำให้เรียนรู้ได้ง่าย
  - Cross platform
  - เป็น Web base (thin-client)
  - OOP (Object Oriented Programming)

1996 Java 1.0

- ได้มีการพัฒนา Tool kit สำหรับการพัฒนาโปรแกรม JDK1.1
- Java Bean, Java(TM)
- IBM, Netscape, Sun จับมือกันหนุน Java

จุดมุ่งหมายของ Java

- **Simple**
  - ไม่ซับซ้อน มีพื้นฐานจาก C,C++
  - เพิ่มความสามาระที่ Compiler
  - ตัด preprocessor command, function prototype และ header file
  - ตัด structure, unions, bit fields และ enumerated type, typedef
  - สามารถสร้างโปรแกรมเกี่ยวกับ GUI, multitasking, network, plug-in ได้ง่าย

จุดมุ่งหมายของ Java

- **Handle Error**
  - exception handling
  - ไม่ให้ใช้ pointer อ้างถึง memory โดยตรง จะมี garbage collector ช่วยในการจัดการคืน memory ที่ใช้แล้วให้กับระบบโดยอัตโนมัติ
  - type checking

จุดมุ่งหมายของ Java

- **Secure แบ่งออกเป็น 3 ระดับ**
  - **ระดับล่าง** ไม่อนุญาติให้ direct access memory ผ่าน pointer
  - **ระดับกลาง** bytecode-verifyer (ตรวจสอบรหัสคำสั่ง)
  - **ระดับสูง** sandbox model โปรแกรมที่ถูก download จากเครื่องอื่นถือว่าเป็น Untrusted codes จะถูกเก็บใน sand box ควบคุมด้วย Security Manager เช่น read/write permition

SmallTalk

- เป็นต้นแบบของ virtual machine
- interpretation แปลโปรแกรมทีละบรรทัด เป็นภาษาสมมติ และถูกทำงานโดยเครื่องจักรสมมติ
- ทำงานบนเครื่องคอมพิวเตอร์ใดๆ ก็ได้ที่มี Smalltalk interpreter
- ข้อเสียคือ โปรแกรมทำงานช้ามาก

JVM (Java's Virtual Machine)

- แปลภาษาโดย compiler
- ทำงานโดย interpreter
- source code เป็นไฟล์ .java
- เมื่อ compile แล้วจะได้เป็น .class ซึ่งประกอบด้วยคำสั่งของ JVM
- JVM เป็นได้ทั้ง Hardware และ Software และมีการกำหนดมาตรฐานในการสร้าง
- JVM จะทำการแปลงการประมวลผล เป็น nativecode
- opcode ของ JVM มีขนาด 1 byte เท่ากันหมดจึงเรียกว่า byte code

JIT (Just In Time)

- เปลี่ยน Java Class เป็น native code
- โปรแกรมทำงานเร็วขึ้น
- คำสั่งที่ถูกทำซ้ำบ่อยๆ จะทำการแปลงครั้งเดียว ไม่ต้องแปลงทุกครั้ง

JAVA Platform

- **Platform** คือ สภาวะแวดล้อมของ Hardware หรือ Software ซึ่งโปรแกรมทำงานอยู่ Java platform จะเป็นเพียง software only platform เท่านั้น ซึ่งจะทำงานบน platform อื่นอีกต่อหนึ่ง
- Platform ของ Java แบ่งออกเป็น 2 ส่วน
  - **Java Virtual Machine** เปรียบเสมือนระบบปฏิบัติการตัวหนึ่งวิ่งอยู่บนระบบปฏิบัติการอื่นๆ โดยมีโปรแกรม Java (Java bytecode) วิ่งอยู่บน Java VM
  - **Java API** คือ กลุ่มขององค์ประกอบ Software ซึ่งได้สร้างขึ้นไว้เรียบร้อยแล้ว ซึ่งจะช่วยในงานต่างๆ ของโปรแกรม Java เช่น GUI

Core API ที่ Java สนับสนุน

- สิ่งจำเป็นทั่วไป เช่น Object, String, thread, number, I/O, data structure, system property, date&time และอื่นๆ
- Applet
- Networking URL,TCP, UDP และ IP address
- Internationalization ช่วยในการเขียนโปรแกรม ให้สามารถใช้งานในท้องถิ่นสำหรับผู้ใช้ทั่วโลกได้ โปรแกรมจะเปลี่ยนแปลงภาษาให้เข้ากับพื้นที่เฉพาะนั้นๆ
- Security ทุกระดับ electronic license, publickey/privetkey,management,authentication

Core API ที่ Java สนับสนุน (ต่อ)

- ส่วนประกอบของโปรแกรม (Java Bean)
- Object Serialization ทำให้สามารถส่ง Object ข้าม network และ Save Object ได้
- Java Database Connectivity (JDBC) จัดการการเข้าถึง DB
- Standard Extension API ซึ่งใช้ในการ 3D, Server, collaboration, telephony, speech, animation และอื่นๆ

JDK (Java Developer Kit)

- สามารถหา download ได้จาก www.javasoft.com
- รุ่นแรก เป็น JDK 1.0
- JDK1.1 (1997) more standardclass, javabeans
- JDK1.1x (1997-1998) มีการปรับเปลี่ยนอย่างต่อเนื่อง
- JDK1.2 (1998-1999) swing, JFC, JDBC 2.0, Communication
- JDK1.3 (2000) servlet, jsp, …. J2EE
- JDK1.4 (2001) multi language

Java 2(TM) Series

- **J2SE** Standard Edition for develop application
- **J2EE** Enterprise Edition for develop Enterprise application
- **J2ME** Micro Edition for develop mobile, PDA, Palm application

Tools ใน JDK

- Basic Tools
- Remote Method Invocation (RMI) Tools
- Internationalization Tools
- Security Tools
- Java IDL Tools

ในกรณีนี้เราจะศึกษาเฉพาะ Basic Tools เท่านั้น

javac (Java Compile)

javac [options] [source files] [[@files]

options คือ command-line options

source files คือ ไฟล์นามสกุล .java ตั้งแต่ 1 ไฟล์ขึ้นไปที่จะทำการ compile

@files คือ ไฟล์ที่เก็บรายชื่อไฟล์นามสกุล .java ตั้งแต่ 1 ไฟล์ขึ้นไป

java (run java)

java [options] class [argument…]

java [options] -jar file.jar [argument…]

javaw [options] class [argument…]

javaw [options] -jar file.jar [argument…]

option คือ command-line options

class คือ ชื่อของ class ที่จะเรียกใช้งาน

file.jar คือ ชื่อของ java archive file ที่จะเรียกใช้งาน เรียกโดยใช้ -jar

argument คือ ข้อมูลที่ต้องการส่งเข้าไปในโปรแกรมโดยผ่านทางฟังก์ชั่น main

applet viewer

ใช้ run และ debug applet โดยไม่ต้องใช้ web browser

appletviewer [options] urls….

Option คือ command-line options

urls คือ url ที่เก็บชื่อของ applet ไว้

Java Document

javadoc [options] [packagenames] [sourcefiles] [classnames] [@files]

option คือ command-line options

packagenames คือ ชื่อของ package (directory) ที่เก็บ files

sourcefiles คือ file java

class คือ ชื่อของ class ที่จะเรียกใช้งาน

* สามารถเลือกใช้ ตัวใดตัวหนึ่งก็ได้ เช่น file Hello.class

ใช้ javadoc Hello

การสร้าง Document แบบมืออาชีพ

- สามารถเพิ่มคำอธิบายโดยใช้เครื่องหมาย /** */ คร่อมข้อความที่ต้องการแสดง ในแต่ละบรรทัดให้มีเครื่องหมาย * **นำหน้าเสมอ** เช่น

```
/**
* Hello Test
* Test document
*/
```

- คำอธิบายจะ**อยู่เหนือ method หรือ class** ที่ต้องการอธิบายเสมอ

การสร้าง Document แบบมืออาชีพ

- สามารถใช้ Tag ของ HTML ร่วมได้ รวมไปถึงการใช้ CSS (Style Sheet)
- สามารถใช้ Tag พิเศษของ javadoc ร่วมได้

@author	คำอ้างอิงถึงผู้เขียน
@version	เพิ่มคำอ้างอิงเวอร์ชัน
@see	เพิ่ม link ในส่วนของ see also
@param	คำอธิบายเกี่ยวกับ parameter เพิ่มเติม มักใช้กับ method
@return	ระบุค่าของ method ที่ส่งกลับ มักใช้กับ method
@exception	ระบุ exception ที่ throw ของคลาส