

Unit 18

Java Language

Object Oriented Programming for the Internet

Takayuki Dan Kimura

- * Characteristics
- * Goals
- * No More
- * Sample Programs

Characteristics

- Oak Language 1991 James Gosling (Sun)
 OO Language better than C++
- Java Language 1995 \$5M project
 OO + Internet
- Inherited from
 - * Concurrency - Mesa (a research language from Xerox), Monitor
 - * Exceptions - Modula-3
 - * Dynamic linking - LISP
 - * Interface definition - Objective C
 - * Automatic memory management - LISP
 - * Objects and classes - C++
 - * Module - Modula-2, Ada
 - * Ordinary statement - C
- Java = C++ - { bad features } + module + concurrency
- Supporting class libraries for distributed computing (Internet)

Goals

- Powerful yet simple : object oriented
- Faster development : interpreted
- Architecture Neutral : byte code
- Portable : byte code
- Robust : memory management
- High performance : multithreading
- Adaptable : dynamic download (linking)
- Secure : powerful run-time system

No More

- Typedefs, Defines, or Processor
- Structures or Unions
- Functions
- Multiple Inheritance
- Goto Statements
- Operator Overloading
- Automatic Coercions
- **Pointers !!!**

Sample Programs

Hello World

```
import java.applet.Applet;  
import java.awt.Graphics;  
  
public class HelloWorld extends Applet {  
  
    public void paint(Graphics g) {  
  
        g.drawString("Hello world!", 50, 25);  
  
    }  
}
```

Data Structure

```
class Stack {  
  
    static final int    STACK_EMPTY = -1;  
  
    Object[] stackelements;  
  
    int topelement = STACK_EMPTY;  
    ...  
    boolean isEmpty() {  
        if (topelement == STACK_EMPTY)  
            return true;  
        else  
            return false;  
    }  
  
    Object pop() {  
        if (topelement == STACK_EMPTY)  
            return null;  
        else {  
            return stackelements[topelement--];  
        }  
    }  
  
}
```

Threading (Clock Demo)

```
import java.awt.Graphics;
import java.util.Date;

public class Clock extends java.applet.Applet implements Runnable {

    Thread clockThread = null;

    public void start() {
        if (clockThread == null) {
            clockThread = new Thread(this, "Clock");
            clockThread.start();
        }
    }

    public void run() {
        // loop terminates when clockThread is set to null in stop()
        while (Thread.currentThread() == clockThread) {
            repaint();
            try {
                clockThread.sleep(1000);
            } catch (InterruptedException e){
            }
        }
    }

    public void paint(Graphics g) {
        Date now = new Date();
        g.drawString(now.getHours() + ":" + now.getMinutes() + ":" +
now.getSeconds(), 5, 10);
    }

    public void stop() {
        clockThread = null;
    }
}
```