Introduction to Java

Object-Oriented Programming
Outline

- History of Java
- Basic features
- Java applications & Java applets
- Hello World!

Readings
- Core Java 2, chapter 1, 2, 3
History

- 1991: developed by Sun Microsystems as a small programming language for embedded household devices
  - Initially called Oak
- 1995: Java 1.0 released
  - “Write Once, Run Anywhere”
  - Became popular with webpages running applets
- Nowadays, widely accepted as a multi-purpose programming language/technology
  - Portable, safe
  - Object-oriented, component-oriented
Compiling and interpreting

- Program source code compiled into bytecode
- Bytecode is executed in an interpreter environment (virtual machine)
Why Java?

- Easy to use
  - Addresses the weaknesses of older programming languages
  - Object-oriented
  - Supports good programming styles
- Interpreter environment
  - Increased portability
  - Safe
    - Branches *always* to valid locations
    - Data *always* initialized, references *always* type-safe
    - Access to "private" or "package private" data and methods is rigidly controlled.
- Features
  - Built-in multi-threading
Types of Java applications

- Desktop application - J2SE
  - Java Application: normal Java application running on desktops
  - Java Applet: embedded application running within Web browsers

- Server application - J2EE
  - JSP và Servlet

- Mobile (embedded) application – Java ME (J2ME – Java 2 Microedition)
Java features

- JVM – Java virtual machine
- Automatic gabbage collection
- Safe programs
JVM – Java Virtual Machines

- Virtual machines depend on specific platforms (hardware, OS)
- Provide Java programs with (platform-independent) run-time environments
- Ensure system security
- Normally provided as software
  - JRE - Java Runtime Environment
- Java platform: JVM + APIs
Garbage Collection

- Java provides a system-level process that monitors memory allocations
- Garbage Collection
  - Mark and free no-longer-used memory
  - Automatic
  - Specific mechanisms depend on versions of virtual machine
JDK – Java Development Kit

- Free development and run-time environment provided by Sun Microsystems (http://java.sun.com)
  - Most widely used Java software development kit
- Main components:
  - `javac` compiler, converts source code into Java bytecode
  - `java` interpreter and application loader
  - `appletviewer` interpreter, run and debug Java applets without a web browser
  - `javadoc` documentation generator, automatically generates documentation from source code comments
  - `jdb` debugger
  - `javap` class file disassembler
JIT
Just-In-Time Code Generator
Java Applications

- Complete application programs
- Command-line or GUI
- Launched by java command
Hello World application

TestGreeting.java:

```java
public class TestGreeting{
    public static void main (String[] args) {
        System.out.println("Hello, world");
    }
}
```

same name with class
public class
public static method
public class TestGreeting{
    public static void main (String[] args) {
        System.out.println("Hello, world");
    }
}
package
object
message
Compile and run

- Compile TestGreeting.java
  javac TestGreeting.java
- Run
  java TestGreeting
- Result
  Hello, world
Small improvement

Two classes in separated files

**TestGreeting.java:**
```java
public class TestGreeting {
    public static void main(String[] args) {
        Greeting gr = new Greeting();
        gr.greet();
    }
}
```

**Greeting.java:**
```java
public class Greeting {
    public void greet() {
        System.out.print("Hello, world");
    }
}
```
Compile and run

- Compile TestGreeting.java
  javac TestGreeting.java
    Greeting.java is automatically compiled

- Run
  java TestGreeting

- Result
  Hello, world
Java Applets

- Embedded into webpages, i.e. run in web browsers.
  - Or in appletviewer
- Limited graphical interface
- No access to client’s resources
  - can do no evil
A simple applet

Welcome.java:

// Java packages
import java.awt.Graphics;
import java.applet.Applet;

public class Welcome extends Applet {

    public void paint(Graphics g) {
        // call superclass version of method paint
        super.paint(g);

        // draw a String
        g.drawString("Welcome to Java programming!", 25, 25);
    }
}
Embedded into a Webpage

Welcome.html:

```html
<html>
<applet code = "Welcome.class"
    width = "300" height = "45">
</applet>
</html>
```
Running (in Web browser)
Run

appletviewer Welcome.html
Homework

http://coltech.vnu.edu.vn/~chauttm/oop2010f