GUI Design and Programming

Part - I

GUI realization using Java

Lecturer: Dr. Daniel Pak-Kong LUN
Room: DE608 Tel: 27666255
E-Mail: enpklun@polyu.edu.hk
References:


### Contents

#### Getting Started
- What is Java?
- A Taste of Java
- Programming Environments

#### Learning the Java language
- Object Oriented Programming
- The Nuts and Bolts
- Objects and Classes in Java

#### Java User Interface
- Overview: The AWT Package
- Visual GUI Builder
- Working Principles
- Working with Sound, Graphics and Images

#### Advanced Topics
- Threads
- Animation
Getting Started

1.1 What is Java?
Java is two things:

Programming language and Platform

- Simple
- Object-oriented
- Distributed
- Interpreted
- Robust
- Secure
- Architecture-neutral
- Portable
- High-performance
- Multithreaded
- Dynamic
Java programs are both compiled and interpreted.
Java bytecodes

- Allow Java programs to be **written once but running on almost any platform**
Platform: A hardware or software environment in which a program runs

The Java Platform:

Software ONLY platform runs on other hardware-based platform

Java Platform has two components
Java Virtual Machine (VM):
• Mechanism which *seems to execute the bytecodes* e.g. Interpreter
• Virtual because the actual execution is *done by the hardware, not the VM*

Java Application Program Interface (API):
• A large collection of *ready-made software components*, e.g. graphic, network, etc.
• Group into *libraries* (packages) of related components
Java Programs:

Applet:
- A special kind of Java program that runs on a Java-enabled browser, e.g. Netscape, IE
- Must follow a programming convention

Application:
- A standalone program that runs directly on the Java platform
- Similar to other languages, e.g. C, C++
Why Java???

- Get started quickly
- Write less code
- Write better code
- Develop programs faster
- Avoid platform dependencies with 100% pure Java
- Write once, run anywhere
- Distribute software more easily
Getting Started

1.2 A taste of Java
Example 1: The “Hello World” Application

**Step 1: Create a Java Source File**

Using a text editor, create a file named `HelloWorldApp.java` with the following Java code:

```java
/**
 * The HelloWorldApp class implements an application that
 * simply displays "Hello World!" to the standard output.
 */

class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");  //Display the string.
    }
}
```
Step 2: Compile the Source File
a. Open a DOS prompt window in your PC
b. Type
   javac HelloWorldApp.java

• If succeed, the compiler creates a file named
  HelloWorldApp.class in the same directory (folder) as the Java source file (HelloWorldApp.java)
• This class file contains Java bytecodes
• If fail, make sure the program is typed and named exactly as shown above, using the capitalization shown
Step 3: Execute the Bytecodes

Type

```
java HelloWorldApp
```

Hello World!
Example 2: The “Hello World” Applet

Step 1: Create a Java Source File
Create a file named HelloWorld.java with the Java code shown here:

```java
import java.applet.Applet;
import java.awt.Graphics;

public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 50, 25);
    }
}
```
Step 2: Compile the Source File
a. Open a DOS prompt window in your PC
b. Type
   ```
   javac HelloWorld.java
   ```

- If succeed, the compiler creates a file named `HelloWorld.class` in the same directory (folder) as the Java source file (`HelloWorld.java`)
- This class file contains Java bytecodes
- If fails, make sure you typed in and named the program exactly as shown above, using the capitalization shown
Step 3: Create an HTML file that includes the Applet

Using a text editor, create a file named `Hello.html` in the same directory that contains `HelloWorld.class`. This HTML file should contain the following text:

```html
<HTML>
<HEAD>
<TITLE> A Simple Program </TITLE>
</HEAD>
<BODY>

Here is the output of my program:
<APPLET CODE="HelloWorld.class" WIDTH=150 HEIGHT=25>
</APPLET>
</BODY>
</HTML>
```
Step 4: Run the Applet

To run the applet, you need to load the file Hello.html into an application that can run Java applets.

This application might be a Java-compatible browser (e.g. Netscape) or another Java applet viewing program (e.g. Applet Viewer).
Getting Started

1.3 Programming Environments
Many programming tools are available in the market

**JDK - Java Development Kit**
- Command-line based tool
- No GUI support. Have to develop Java program in DOS mode

**Java Workshop, Visual Café for Java, JBuilder, Visual Age for Java**
- Windows based tool
- With GUI support and visual programming capability

**Java Studio**
- High level Java programming tool
- Programming by connecting objects graphically
Java Workshop 2.0

Provide
- Intuitive user interface
- JavaBeans support
- Visual Java GUI Builder
- Profiler
- Support for JDK 1.1 standards
- Multi-platform support
Project

- Project is the fundamental unit of job in Java Workshop 2.0
- A project keeps track of the various files needed to build, browse, run, and debug
- Projects are grouped together into portfolios
- There are four different project types:
  - Applet
  - Standalone Program
  - Java package - a library of classes
  - Java beans - reusable software components
**Java WorkShop Tools**

**BUILD**
Build the current project

**RUN**
Run the current project
- For applet, an HTML file is created and run in the applet viewer
- For standalone program, the output will go to the console

**PROFILE**
Produce a performance report
- Showing how much time is spent in each program method
- Showing how many times each method is executed

**SEARCH**
Find references to strings of text in specified source files
Java WorkShop Tools

**PROJECT MANAGER**
Create, import, edit, and remove projects

**ATTRIBUTE**
Edit properties of interface components in GUI builder projects

**HELP**
Provide information about Java WorkShop
Project Development Stages

- Create Project
- Edit Source
- Build Project
- Debug Project
- Test/Profile Project
- Browse Source
- Edit Project
Example 3: To build a project with Java Workshop 2.0

An Applet called Blink is to be built and run

Step 1: Create project

1. Click the Projects button
2. Set the current portfolio to be personal
3. Choose File -> New -> Project
4. Type Blink in the project name box
5. Click the Applet radio button for the application type
6. Click the Projects button
7. Click the No GUI radio button
8. Save the Blink project in the default directory
9. Add existing Java files to the Blink project
10. Use the html page generated by Java WorkShop
Step 2: Build project

1. Check the Java WorkShop Project Manager window for the name of the current project to be Blink
2. Click the Build button on the Java WorkShop main tool
Step 3: Run project

1. Check the Java WorkShop Project Manager window for the name of the current project to be Blink
2. Click the Run button on the Java WorkShop main tool
Exercise:

Complete Tutorials 1, 2 and 3 in the Java WorkShop Tutorial

If you have time, complete also Tutorial 7