Introduction to Java

Chaiwoot Boonyasiriwat

February 11, 2015

Outline

- History of Java
- Hello World and Java Compilers
- Object-oriented programming (OOP)
- Your first class
- Command line program
- Windows program
- Graphic user interface (GUI)
- Drawing Graphics
- Animation
- Applet

History of Java

- Java was originally developed by Sun Microsystem for embedded systems in 1991.
- Java was first released in 1995 for web applications.
- Java was based on C++.
- Java is an object-oriented programming (OOP) language.
- Java was named after Java coffee beans of Indonesia. It was originally named “oak”.
- Java is one of the most widely used languages.

Hello World

HelloWorld.java

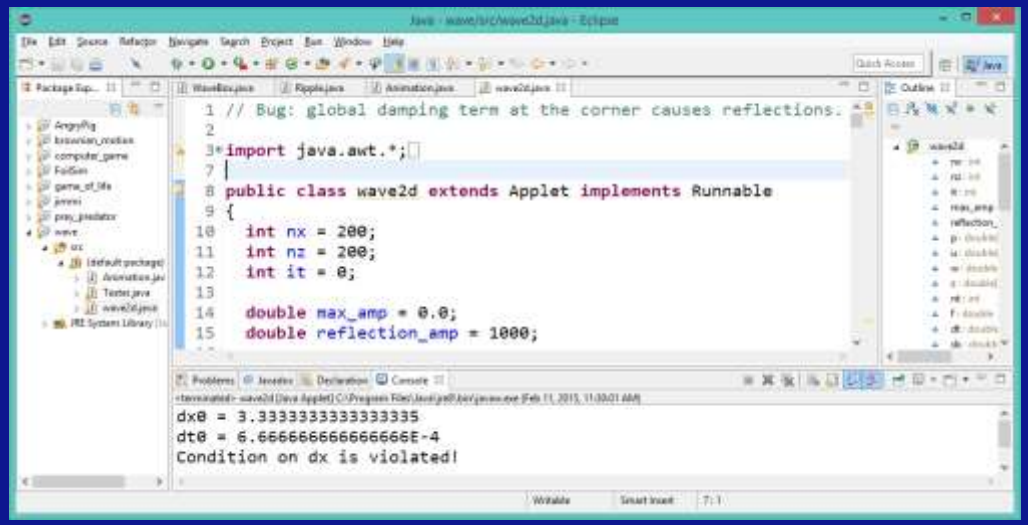
```
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
```

hello.c

```
#include <stdio.h>
void main(int argc, char** args) {
  printf("Hello World!\n");
}
```

Java Compiler and IDE

- Java Developer Kit (JDK) including compiler can be obtained at http://www.oracle.com/technetwork/java/javaseproducts/downloads/index.html
- Integrated Development Environment (IDE) popular for Java programming is Eclipse which can be obtained at https://eclipse.org/downloads/



Hello World in a Window

HelloWorldWindow.java

```
import javax.swing.*;
public class HelloWorldWindow {
  public static void main(String[] args) {
    JOptionPane.showMessageDialog(
      new JFrame(), “Hello World!”, “Test”,
      JOptionPane.INFORMATION_MESSAGE);
  }
}
```

Object-Oriented Programming

- **Encapsulation**: data and methods are combined as **class** which is used to create an **instance** of objects.
- **Inheritance**: parent class's properties can be inherited by subclasses.
- **Polymorphism**: overloading, overriding, dynamic method binding

Encapsulation

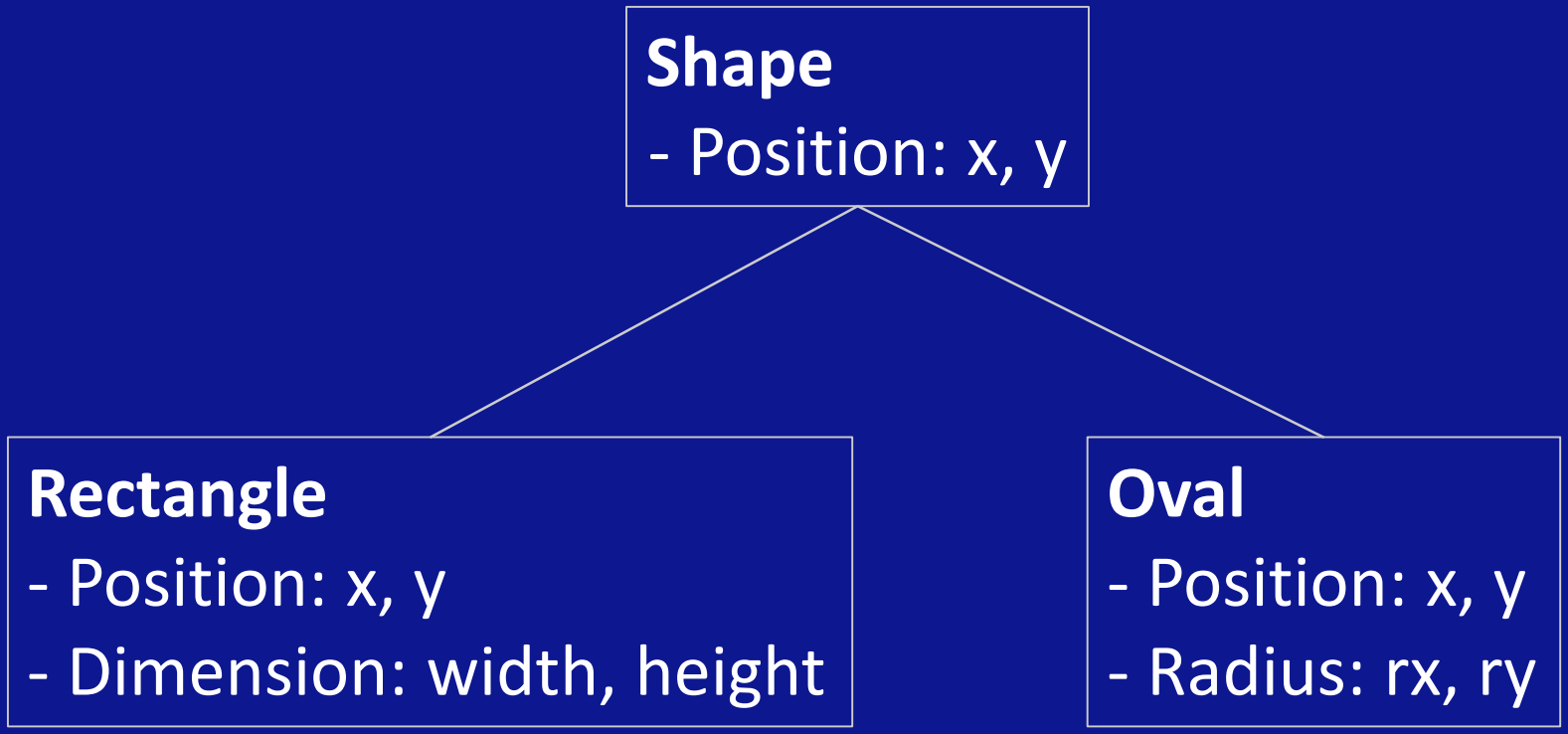
```
// Class declaration
class Rectangle {
   int x, y;  // Data
   Rectangle(int x, int y){ //Constructor
      this.x = x;
      this.y = y;
   }
   int getArea(){ return x*y; } // Method
}
```

Use class constructor to create an instance:

```
Rectangle r = new Rectangle(10,20);
```

Inheritance

Subclass can inherit properties of **parent class**



Inheritance: Example

```
class Shape {
   int x, y;
   Shape(int x, int y) {
      this.x = x; this.y = y;
   }
}
class Oval extends Shape {
   int rx, ry;
   Oval(int x, int y, int rx, int ry){
      super(x,y);
      this.rx = rx; this.ry = ry;
   }
}
```

Inheritance: Example (cont.)

```
class Drawing extends JPanel {
  Oval o1 = new Oval(1,1,10,10);
  Oval o2 = new Oval(1,10,5,5);
  Oval o3 = new Oval(1,15,20,20);
  o1.rx = 20; o1.ry = 20;

  public void paint(Graphics g) {
    g.drawOval(o1.x,o1.y,o1.rx,o1.ry);
    g.drawOval(o2.x,o2.y,o2.rx,o2.ry);
    g.drawOval(o3.x,o3.y,o3.rx,o3.ry);
  }
}
```

Polymorphism

- **Overloaded methods** are methods with the same name signature but either a different number of parameters or different types in the parameter list.
- **Overridden methods** are methods that are redefined within an subclass.
- **Dynamic method binding** is the ability of a program to resolve references to subclass methods at runtime.

Method Overloading

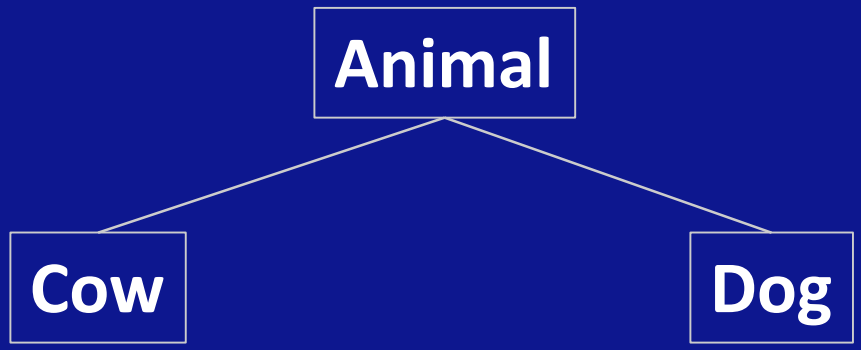
```
int add(int a, int b) {
   return a+b;
}
float add(float a, float b) {
   return a+b;
}
double add(double a, double b) {
   return a+b;
}
```

Method Overriding

A subclass can behave differently from the parent class by overloading the inherited methods.

```
class Car {
   int getSpeed() {
      return 40; // km/s
   }
}
class SportCar extends Car {
   int getSpeed() {
      return 160; // km/s
   }
}
```

Dynamic Method Binding




```
public class AnimalReference
{
   public static void main(String args[]){
      Animal ref;
      Cow aCow = new Cow("Bossy");
      Dog aDog = new Dog("Rover");
      ref = aCow; ref.speak();
      ref = aDog; ref.speak();
   }
}
```

Simple Java Application

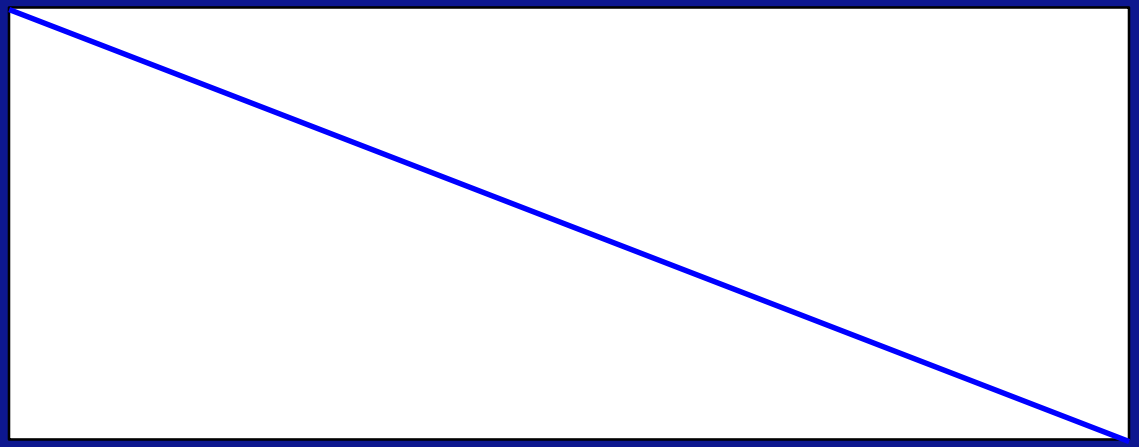
SimpleApp.java

```
import java.awt.*;
public class SimpleApp extends Frame {
  public static void main(String[] args) {
    SimpleApp app = new SimpleApp();
    app.setVisible(true);
  }
  public SimpleApp() {
    setSize(400,200);
    setTitle("Simple Application");
  }
}
```

Drawing Lines

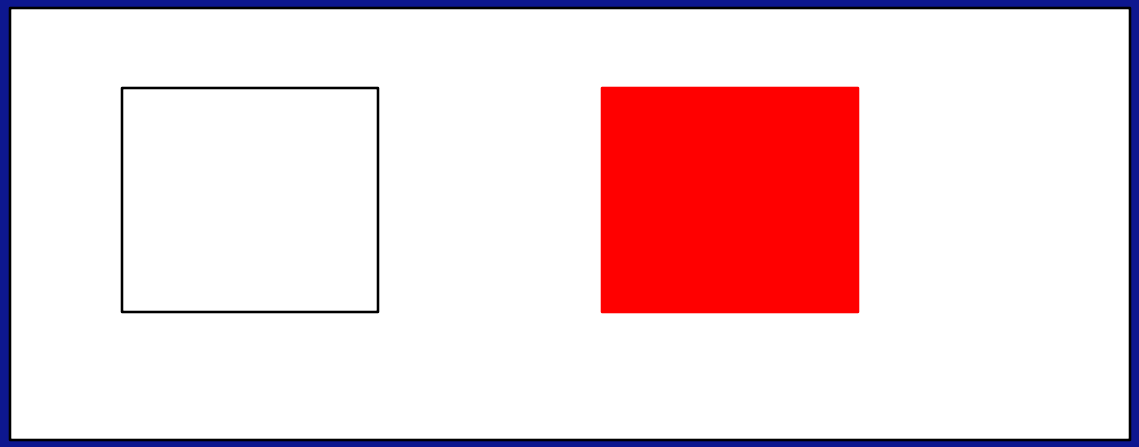
```
import java.awt.*;

public void paint(Graphics g) {
   g.setColor(Color.blue);
   g.drawLine(0,0,100,50);
}
```



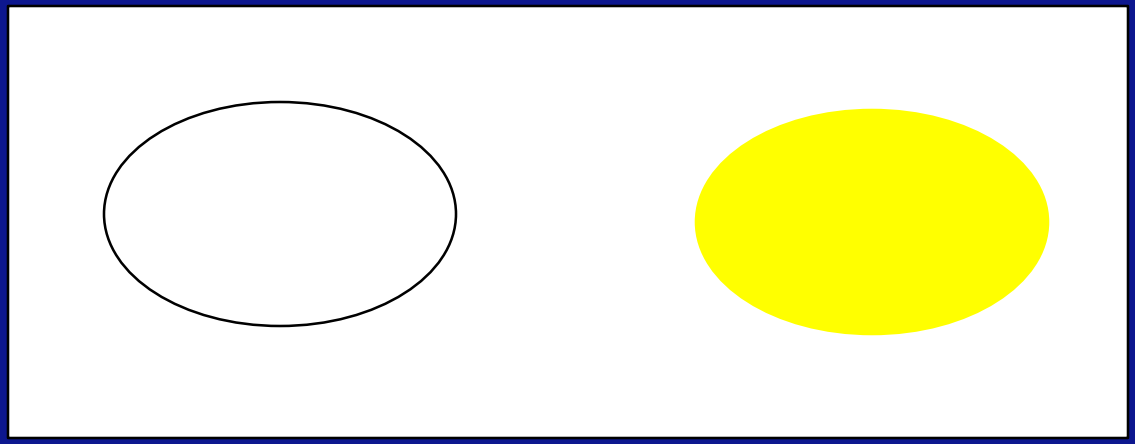
Drawing Rectangles

```
public void paint(Graphics g) {
   setColor(Color.black);
   g.drawRect(x1,y1,w1,h1);
   setColor(Color.red);
   g.fillRect(x2,y2,w2,h2);
}
```



Drawing Ovals

```
public void paint(Graphics g) {
   setColor(Color.black);
   g.drawOval(x,y,rx,ry);
   setColor(Color.yellow);
   g.fillOval(x,y,rx,ry);
}
```



Drawing Texts

```
public void paint(Graphics g) {
   setColor(Color.black);
   g.drawString("Hello Guys",x,y);
}
```

Graphic User Interface

Abstract Windowing Toolkit (AWT)

```
import java.awt.*;
```

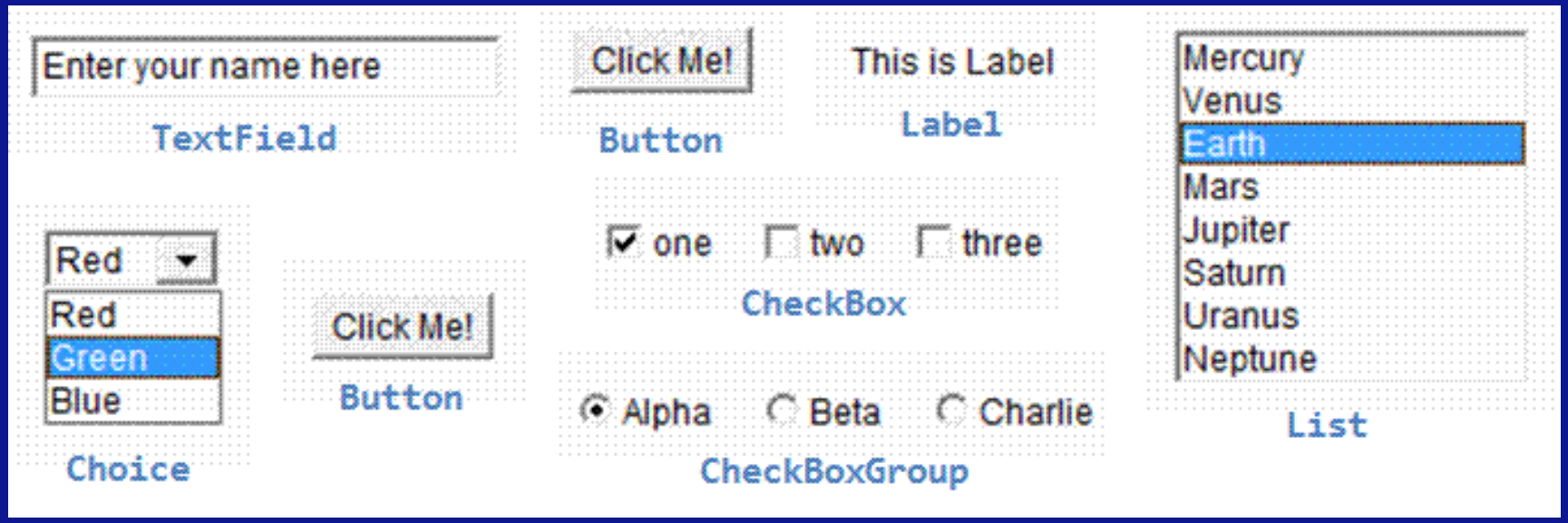
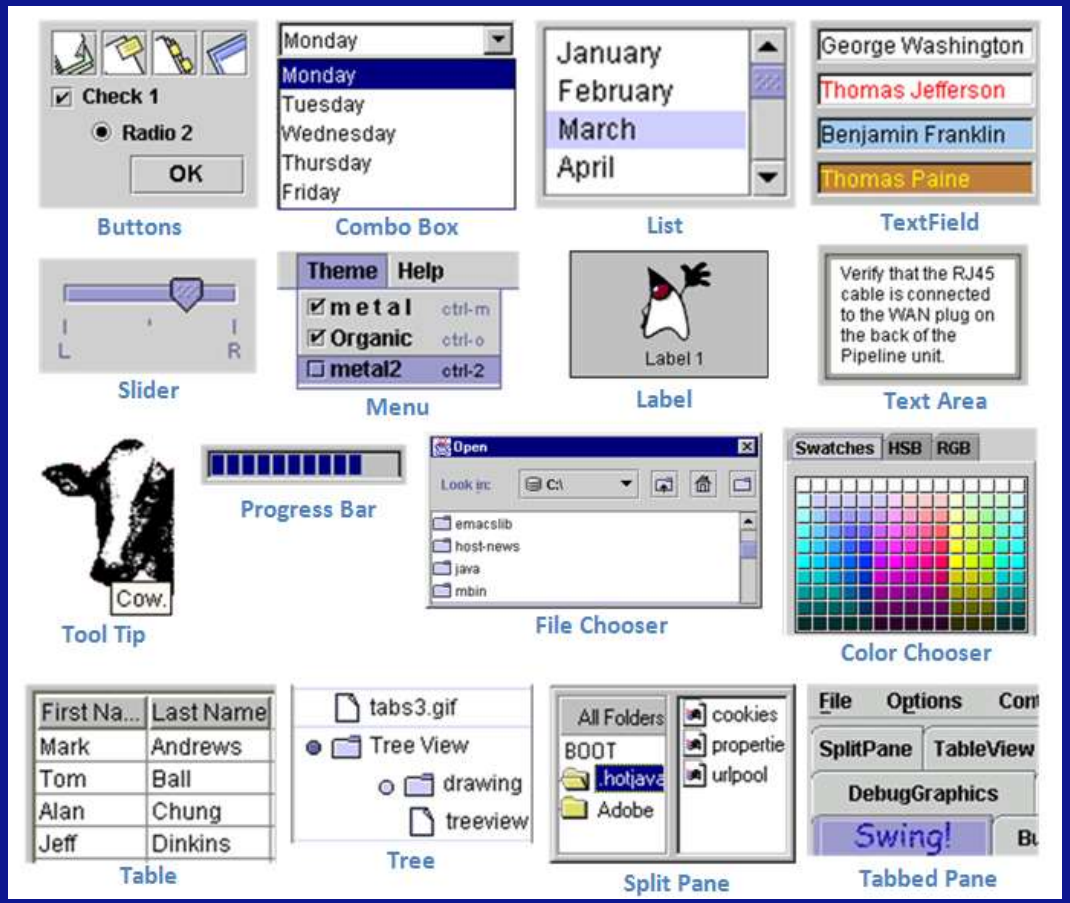


Image source: http://www3.ntu.edu.sg/home/ehchua/programming/java/J4a_GUI.html

Graphic User Interface

Swing Components

```
import javax.swing.*;
```



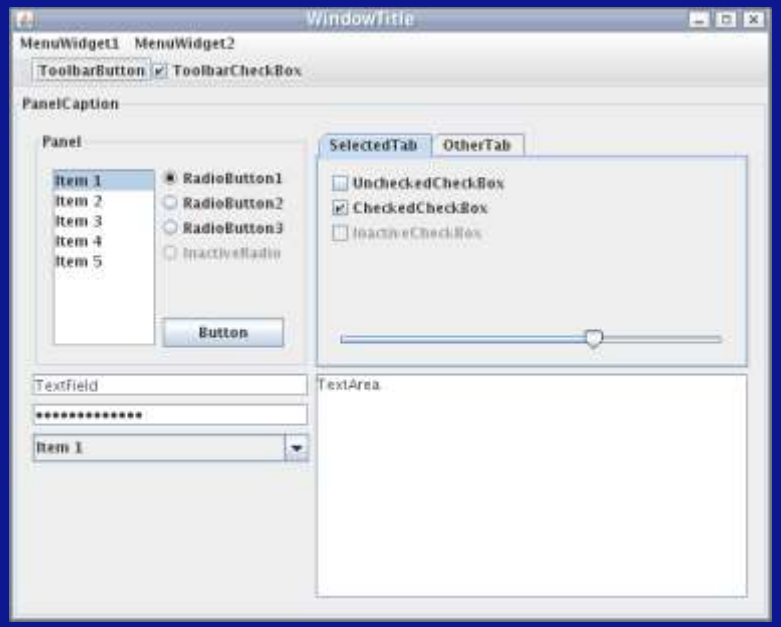


Image sources: https://www3.ntu.edu.sg/home/ehchua/programming/java/images/Swing_Components.png
http://en.wikipedia.org/wiki/Swing_(Java)#mediaviewer/File:Gui-widgets.png

References

Deitel, P., and H. Deitel, 2010, Java How to Program, Eighth Edition, Pearson Education.