

A Bit of Java: from source-code to byte-code

Java Source Code

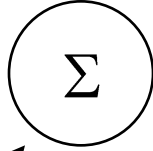
```
FileDemo1_WebApp
File Demo1_WebApp Frame.jsp
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
import javax.swing.*;
import java.awt.*;
import java.net.*;

class FileDemo1 extends JFrame
 implements ActionListener, ActionListener {

    //List all the URIs that will be used in this code
    private Button loadButton;
    private Button loadButton1;
    private Button saveButton;
    private TextFile saveTextField;
    private JLabel saveLabel;

    public static void main (String [] args) {
        if (args.length < 1) {
            //Help commented with the filename
            System.out.println ("FileDemo1: JIP address of server?");
            return;
        }
        File file = new File (args [0]);
        if (!file.exists ())
            System.out.println ("File does not exist");
        else
            System.out.println ("File exists");
    }
}
FileDemo1.java
```

Java Compiler



Java Bytecode

```
FileDemo1_WebApp
File Demo1_WebApp Frame.jsp
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
class FileDemo1 {
    private Button loadButton;
    private Button loadButton1;
    private Button saveButton;
    private TextFile saveTextField;
    private JLabel saveLabel;

    public static void main (String [] args) {
        if (args.length < 1) {
            //Help commented with the filename
            System.out.println ("FileDemo1: JIP address of server?");
            return;
        }
        File file = new File (args [0]);
        if (!file.exists ())
            System.out.println ("File does not exist");
        else
            System.out.println ("File exists");
    }
}
FileDemo1_WebApp.class
```

Additional Classes

```
FileDemo1_WebApp
File Demo1_WebApp Frame.jsp
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
import javax.swing.*;
import java.awt.*;
import java.net.*;

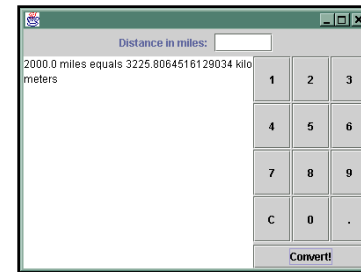
class FileDemo1 extends JFrame
 implements ActionListener, ActionListener {

    //List all the URIs that will be used in this code
    private Button loadButton;
    private Button loadButton1;
    private Button saveButton;
    private TextFile saveTextField;
    private JLabel saveLabel;

    public static void main (String [] args) {
        if (args.length < 1) {
            //Help commented with the filename
            System.out.println ("FileDemo1: JIP address of server?");
            return;
        }
        File file = new File (args [0]);
        if (!file.exists ())
            System.out.println ("File does not exist");
        else
            System.out.println ("File exists");
    }
}
FileDemo1_WebApp.class
```

Java Virtual Machine

JVM



Application or Applet

Java Vocabulary

- Object - A “real-world” entity that is modeled in software.
- Method - essentially a function **called** to modify an object’s behavior, appearance, or content.
 - Balloon.changeColor(“yellow”);
- Constructor - A specific method used to create a new “instance” of an object.
- Class - a software template comprised of the data and methods **associated** with object creation and manipulation.
- Instantiation - the process of creating an “instance” of an object.
 - Ex. myBalloon = new Balloon(“green”, 10, 10);
- Inheritance - the acquisition of software traits by a class (child) from its super-class (parent).
 - Ex. public class myApplet extends Applet
- Polymorphism - from Greek *having many forms*. Within a hierarchy of classes -- I.e. inheritance -- a method with a **common** signature (name) but different scopes (range of influence) **is referred to as polymorphic**.
 - Ex1. class plane { public void flying() {System.out.println(“See the plane fly.”);} }
 - Ex2. class wing { public void flying() {System.out.println(“The wing provides lift.”);} }