WRITING APPLETS:

An applet is a Java program written in a special format to have a graphical user interface, which allows a user to interact with a program by keyboard and mouse, or by performing other familiar actions.

The basic structure of a Java applet:

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class AppletName extends Applet
{
}
```

THE IMPORT STATEMENTS:

import java.applet.Applet; provides the applet class.

import java.awt.*; provides button class, textbox class etc. it is collection of classes for drawing GUI items.

import java.awt.event.*; provides ActionEvent, ActionListener etc.

THE CLASS DEFINITION:

public class AppletName extends Applet

Here class inherited Applet class, because some methods are taken from the class Applet. By using import java.applet.Applet; statement, Applet class was imported.

Related Posts:

1. Can Java have same name variable
2. Types of variables in Java programming
3. JAVA and its Support Systems
4. JAVA environment

5. JAVA program structure

6. Tokens

7. Java statements

8. Java virtual machine

9. C++ Versus JAVA

10. Constants and Variables in Java

11. Data types JAVA

12. Defining a class

13. Constructor in JAVA

14. Array in Java

15. Applet

16. Applets Vs Applications

17. Applets life cycle

18. Creating an Executable Applet

19. Graphics in Applet

20. Applet image display

21. Applet digital clock

22. Applet mouse event handling

23. JDBC

24. Execute an SQL Statement

25. Process the result

26. CLOSE THE DATABASE CONNECTION

27. File handling

28. Define a class to declare an integer array of size n and accept the elements into the array.

29. Define a class to declare an array of size 20 of the double datatype, accept the elements into the array and perform the following: Calculate and print the sum of all

the elements.

30. Java program for String, to uppercase, to equal, length of string
31. Write a Java program for Buble sort.
32. Write a Java program String to uppercase and count words startig with 'A'
33. How to set path in Java
34. Understanding public static void main (String args[]){ } in Java
35. Difference between static and non static methods in Java