In Java, statements are individual instructions or commands that perform a specific action. Each statement typically ends with a semicolon (;) to mark its completion.

Here are some common types of statements in Java:

1. Variable Declaration and Assignment:

```
int age; // Variable declaration
age = 25; // Variable assignment
```

This statement declares an integer variable named age and assigns the value 25 to it.

2. Expression Statements:

```
int result = 10 + 5; // Expression statement
System.out.println("The result is: " + result); // Method invocation
statement
```

Expression statements perform operations or function calls. The first statement adds 10 and 5, and assigns the result to the variable result. The second statement calls the println method to display a message along with the value of result.

3. Control Flow Statements:

- ████████████████████████

```
int number = 7;
if (number % 2 == 0) {
    System.out.println("The number is even.");
} else {
    System.out.println("The number is odd.");
}
```

This statement checks if number is divisible by 2. If the condition is true, it prints "The number is even."; otherwise, it prints "The number is odd."

- ████████████████████████████

```
for (int i = 0; i < 5; i++) {
    System.out.println("Iteration: " + i);
}

int count = 0;
while (count < 3) {
    System.out.println("Count: " + count);
    count++;
}

int value = 0;
do {
    System.out.println("Value: " + value);
    value++;
} while (value < 3);
```

These statements demonstrate different types of loops. The for loop executes a block of code repeatedly for a specified number of iterations. The while loop continues execution until a

certain condition is met. The do-while loop first executes the code block and then checks the condition for further iterations.

4. Jump Statements:

- ███████████████████████

```java
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        break; // Exit the loop when i equals 5
    }
    System.out.println("Value: " + i);
}

- `continue` statement:
```java
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        continue; // Skip the current iteration when i equals 5
    }
    System.out.println("Value: " + i);
}
```

The break statement is used to exit a loop prematurely, while the continue statement skips the current iteration and moves to the next iteration of a loop.

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 virtual machine

8. C++ Versus JAVA

9. Constants and Variables in Java

10. Data types JAVA

11. Defining a class

12. Constructor in JAVA

13. Array in Java

14. Applet

15. Applets Vs Applications

16. Writing applets

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