Break and Continue in Java - Detailed Notes

1. Introduction

In Java, 'break' and 'continue' are important control statements.

- 'break' is used to terminate loops or switch statements immediately.
- 'continue' is used to skip the current iteration and jump to the next loop cycle.

2. break Statement

- Terminates the innermost loop or switch statement immediately.
- Commonly used with for, while, do-while, and switch.

Syntax:

```
break;
```

Example 1: Using break in a for loop

```
public class BreakExample {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            if (i == 5) {
                break;
            }
            System.out.println(i);
        }
    }
}
```

Output:

```
1
2
3
4
```

Example 2: Using break in a while loop

```
public class BreakWhileExample {
    public static void main(String[] args) {
        int i = 1;
        while (i <= 5) {
            if (i == 3) {
                break;
            }
            System.out.println(i);
            i++;
        }
    }
}
```

Output:

1

2

3. continue Statement

- Skips the remaining code for current iteration and proceeds to next iteration.
- Used with for, while, and do-while loops.

Syntax:

```
continue;
```

Example 1: Using continue in a for loop

```
public class ContinueExample {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            if (i == 3) {
                continue;
            }
            System.out.println(i);
        }
    }
}
```

Output:

1

2

```
4

5
```

Example 2: Using continue in a while loop

```
public class ContinueWhileExample {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            i++;
            if (i == 2) {
                continue;
            }
            System.out.println(i);
        }
    }
}
```

Output:

```
1

3

4

5
```

4. break vs continue

Comparison table between break and continue:

Feature	break	continue
Purpose	Exits the loop or switch	Skips current iteration
After Execution	Moves control outside the loop	Moves control to next iteration
Usage	Loops and switch	Only loops

5. Using Labels with break and continue

- Labels allow you to break or continue outer loops, not just inner ones.

Example: break with label

Break and Continue in Java - Detailed Notes

```java
public class BreakLabelExample {
    public static void main(String[] args) {
        outerLoop:
        for (int i = 1; i <= 3; i++) {
            for (int j = 1; j <= 3; j++) {
                if (i == 2 && j == 2) {
                    break outerLoop;
                }
                System.out.println(i + " " + j);
            }
        }
    }
}
```

Output:

```
1 1
1 2
1 3
2 1
```

Example: continue with label

```java
public class ContinueLabelExample {
    public static void main(String[] args) {
        outer:
        for (int i = 1; i <= 2; i++) {
            for (int j = 1; j <= 3; j++) {
                if (j == 2) {
                    continue outer;
                }
                System.out.println(i + " " + j);
            }
        }
    }
}
```

Output:

```
1 1
2 1
```