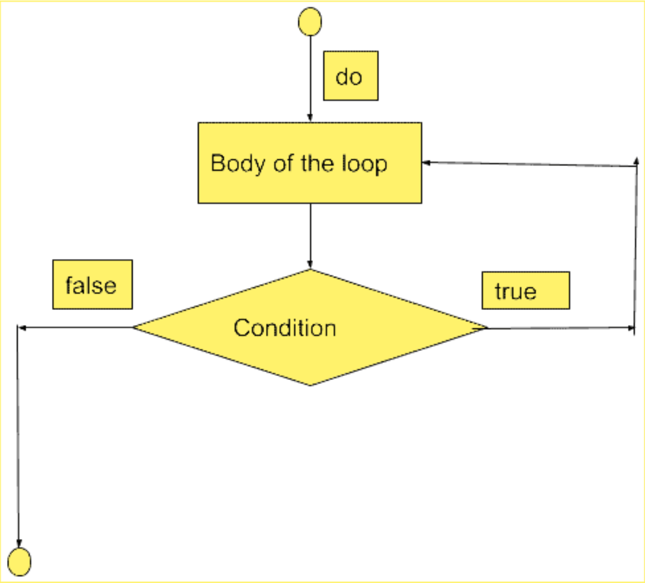
07-Do While Loop

A do-while loop is an exit-controlled control flow structure in programming. It ensures that a block of code is executed at least once and then continues to execute the block as long as a specified condition remains true. The key feature of a do-while loop is that the condition is evaluated after the code block, guaranteeing at least one execution, even if the condition is initially false.

Syntax of Do-While Loop:

```
do {
  // code block to be executed
} while (condition);
```



In this structure, the code block inside the do section is executed first. After execution, the condition in the while statement is evaluated. If the condition evaluates to true, the loop iterates again; if false, the loop terminates.

Example: Do-While Loop Usage

Let's illustrate this with an example where we want to ensure that a message is printed at least once, regardless of the condition.

Example Code:

```
public class Main {
  public static void main(String[] args) {
    int i = 1;
```

```
        do {
            System.out.println(i);
            i++;
        } while (i <= 4);
    }
}
```

In this example:

- The variable i is initialized to 1.
- The do block contains the code to print the value of i and increment it by 1.
- The loop continues to run while i <= 4. As a result, numbers from 1 to 4 are printed.
- Even if i was initially greater than 4, the message would still print once because the condition is checked after the code block execution.

Differences Between While and Do-While Loops

- **Execution Guarantee**: The do-while loop guarantees that the code block is executed at least once, even if the condition is false from the beginning. In contrast, a while loop checks the condition before execution and may not run at all if the condition is false initially.
- **Condition Check**: In a while loop, the condition is checked before the loop body; in a do-while loop, the condition is checked after the loop body.