Last updated: 12 Oct 2021

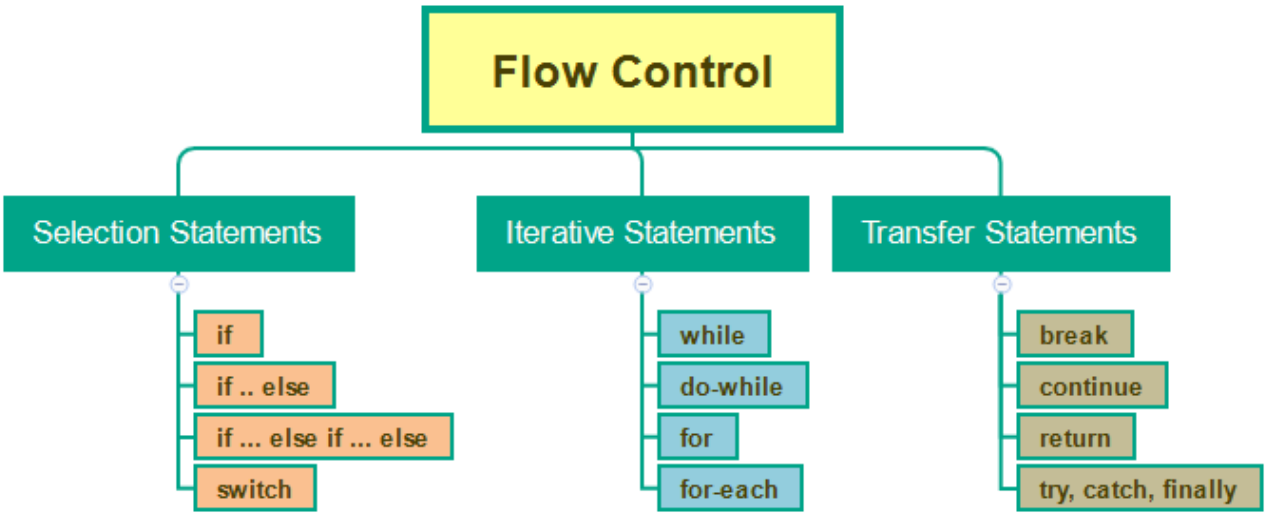
Selenium Java Training - Session 5 - Java (Part 3) - Flow Control Statements

Java (Part 3) - Flow Control Statements

Flow Control

Flow Control describes the order in which statements will be executed at run time.

- There are different types of flow control statements and can be categorized as below:



Selection Statements

Selection Statements have one or more conditions which return either true or false when evaluated.

- Based on the returned value, the set of code will be executed.
    - When condition is true, the set of code will be executed.
    - When condition is false, the set of code wont be executed.

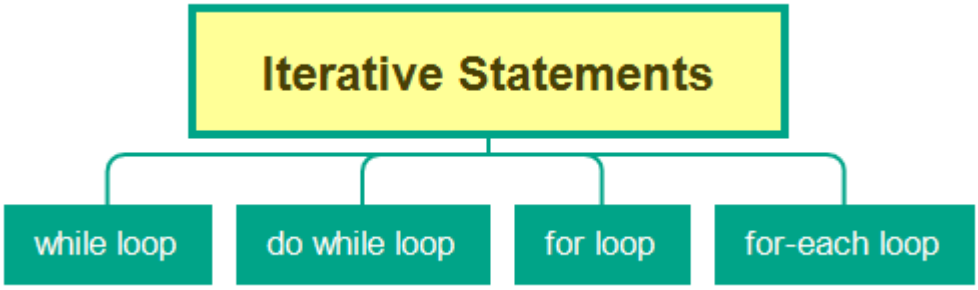
- Different types of Java Selection Statements:

- **if statements**
  - Code inside the **if** decision making statement executed only when the condition provided inside if decision making statement returns true. (View screenshot here)
  - Syntax (View screenshot here)
  - Demonstrate **if** decision making statement
    - Demonstrate the execution of the statements inside if block when the if condition is true. (Demonstrate here)
    - Demonstrate that statements inside if block are skipped from execution when the if condition is false. (Demonstrate here)
- **if ... else statements**
  - While using **if ... else** decision making statement, code inside if block gets executed when the if condition is true and code inside else block gets executed when the if condition is false. (View screenshot here)
  - Syntax (View screenshot here)
  - Demonstrate if else decision making statement
    - Demonstrate the execution of statements inside if block when the if condition is true (Demonstrate here)
    - Demonstrate the execution of statements inside else block when the if condition is false (Demonstrate here)
- **if ... else if ... else statements**
  - **if .. else if .. else** statements contains more than one conditions. If the first if condition returns false, the code inside the if block will be skipped and control will be taken to the next conditions i.e. **else if** conditions. If all the **else if** conditions return false, the code inside else if blocks will be skipped and the code inside the **else** block without condition will be executed. (View screenshot here)
  - Syntax (View screenshot here)
    - Demonstrate the program when all the if and else if conditions have returned false and code inside the else block is executed (Demonstrate here)
    - Demonstrate the program when the if condition has returned true and code inside the if block got executed skipping all the remaining else if and else blocks (Demonstrate here)
    - Demonstrate the program when one of the else if condition has returned true and code inside that else if block got executed skipping all the remaining if, else if and else blocks (Demonstrate here)
- **switch statements**
  - Based on result of a condition expression, switch case chooses one of many possibilities. (View screenshot here)
    - Syntax (View screenshot here)
    - The result of a condition expression needs to result a int or character or a string value.
    - Demonstrate switch case (Demonstrate here)
    - Demonstrate switch case which dont match any case and executes the code in the default section. (Demonstrate here)

Iterative Statements

Iterative Statements helps us in executing the same block of code multiple times.

- Iterative Statements executes the same set of code until the loop condition is satisfied. (View screen-shot here)
- Different types of Iterative Statements:



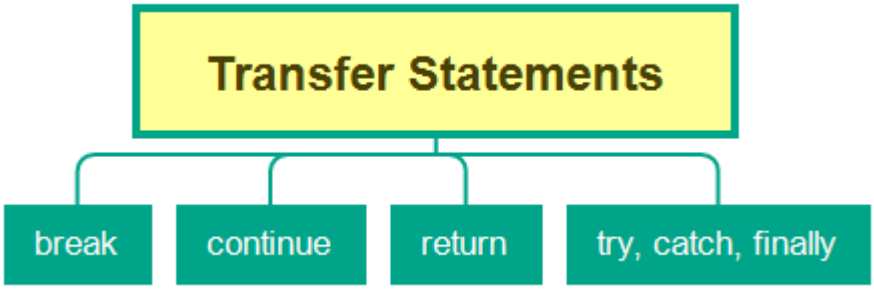
- **while loop**
  - while loop executes the same block of code multiple times i.e. until the boolean condition turns false.
  - while loop tests the condition before executing the code in loop body. (View here)
  - Syntax: View here
  - Demonstrate while loop
    - Demonstrate the while loop when the condition is always true (Demonstrate here)
      - Loop will be iterated infinite times as the condition is always true
    - Demonstrate the while loop when the condition is false (Demonstrate here)
      - Loop wont be executed at-least once as the condition is false.
    - Demonstrate the while loop where the condition is initially true and after few iterations is turned false (Demonstrate here)
      - Loop will be executed until the condition turns false.
- **do while loop**
  - do-while loop works similar to while loop, but the block of code will be executed at-least once even after the condition is false
  - Unlike while loop, do-while loop tests the condition after executing the code in loop body. (View here)
  - Syntax: View here
  - Demonstrate the do-while loop where the condition is initially true and after few iterations has turned false (Demonstrate here)
    - Loop will be iterated multiple times until the condition becomes false
- **for loop**
  - for loop is the most commonly used loop in Java.
  - for loop executes the same block of code multiple times, until the boolean condition turns

    - Loop will be iterated multiple times until the condition becomes false
- **for-each loop**
    - Will be explained later, as it is generally used with Arrays and Collections.

Transfer Statements

Transfer statements are used to transfer the flow of execution from one block of code to a different block of code.

- Different types of Transfer Statements:



- **break;**
    - The purpose of the break; statement is to come out of the statements based on some condition.
    - Demonstrate the usage of break; statements inside any loop statement (Demonstrate here)
- **continue;**
    - The purpose of the continue statement is to skip the current iteration of a loop based on some condition and continue with the next iteration.
    - Demonstrate on how to use continue; statements inside any loop statement (Demonstrate here)
- **return**
    - Will be explained later, as it is used with methods.
- **try, catch, finally**
    - Will be explained later, as it is used in Exception Handling

By,
Arun Motoori