4-Conditionals

Author: Vincent Lau

Learning Objectives

> Understand and use various types of conditionals, if-else & switch

Conditionals

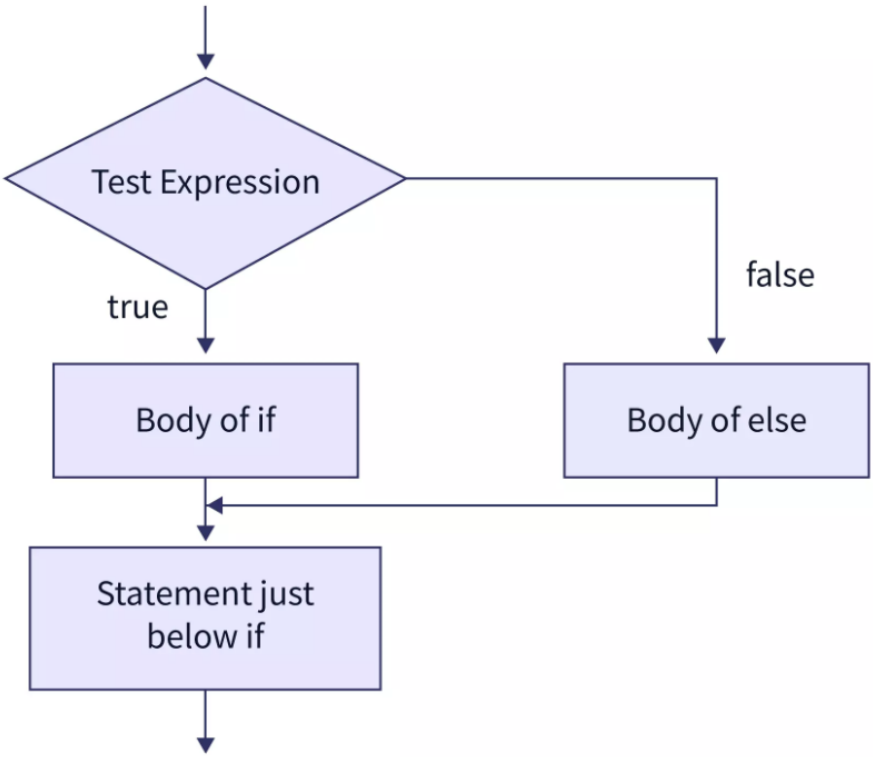
- Conditionals are used for **choosing between two or more paths in a program**. There are two main types of conditionals in Java:
  - `if-else` statements
  - `switch` statements

If-else statements

In Java, the conditional statement `if-else` allows you to **execute different blocks** of code based on certain conditions.

Here's an introduction to the `if-else` statement with examples for beginners to learn:

- The condition is a boolean expression that determines whether the code inside the `if` block or the `else` block is executed.
- If the condition evaluates to `true`, the code inside the `if` block is executed.
- If the condition evaluates to `false`, the code inside the `else` block is executed.




```
if (condition) {
    // Code to be executed if the condition is true
} else {
    // Code to be executed if the condition is false
}
```

Now, let's look at some examples to illustrate the usage of the `if-else` statement:

Example 1: Checking if a number is positive or negative

```
int number = -5;

if (number > 0) {
```

```
    System.out.println("The number is positive.");
} else {
    System.out.println("The number is negative.");
}
// what is printed?
```

In this example, the condition `number > 0` is evaluated.

If it is `true`, the message "The number is positive." is printed. Otherwise, the message "The number is negative." is printed.

Example 2: Determining the maximum of two numbers

```
int a = 10;
int b = 7;
int max = 0;

if (a > b) {
    max = a;
} else {
    max = b;
}
System.out.println("The maximum number is: " + max);
// max = ?
```

In this example, the condition `a > b` is evaluated. If it is `true`, the value of `a` is assigned to `max`. Otherwise, the value of `b` is assigned to `max`.

Example 3: Checking if a number is even or odd

```
int number = 12;

if (number % 2 == 0) {
    System.out.println("The number is even.");
} else {
    System.out.println("The number is odd.");
}
// what is the printed?
```

In this example, the condition `number % 2 == 0` is evaluated.

If it is `true`, the message "The number is even." is printed. Otherwise, the message "The number is odd." is printed.

Example 4: String with if-else

```
String day = "Sunday";

if (day.equals("Sunday")) {
    System.out.println("It's a weekend!");
} else {
    System.out.println("It's a weekday.");
}
// what is printed?
```

In this example, the `if` condition checks if the `day` variable is equal to "Sunday".

If it is, the message "It's a weekend!" is printed. Otherwise, the `else` block is executed, and the message "It's a weekday." is printed.

Example 5: boolean with if-else

```
boolean isRaining = true;

if (isRaining) {
    System.out.println("Remember to take an umbrella!");
} else {
    System.out.println("No need for an umbrella.");
}
```

In this example, the `if` condition checks if the `isRaining` variable is `true`.

If it is, the message "Remember to take an umbrella!" is printed. Otherwise, the `else` block is executed, and the message "No need for an umbrella." is printed.

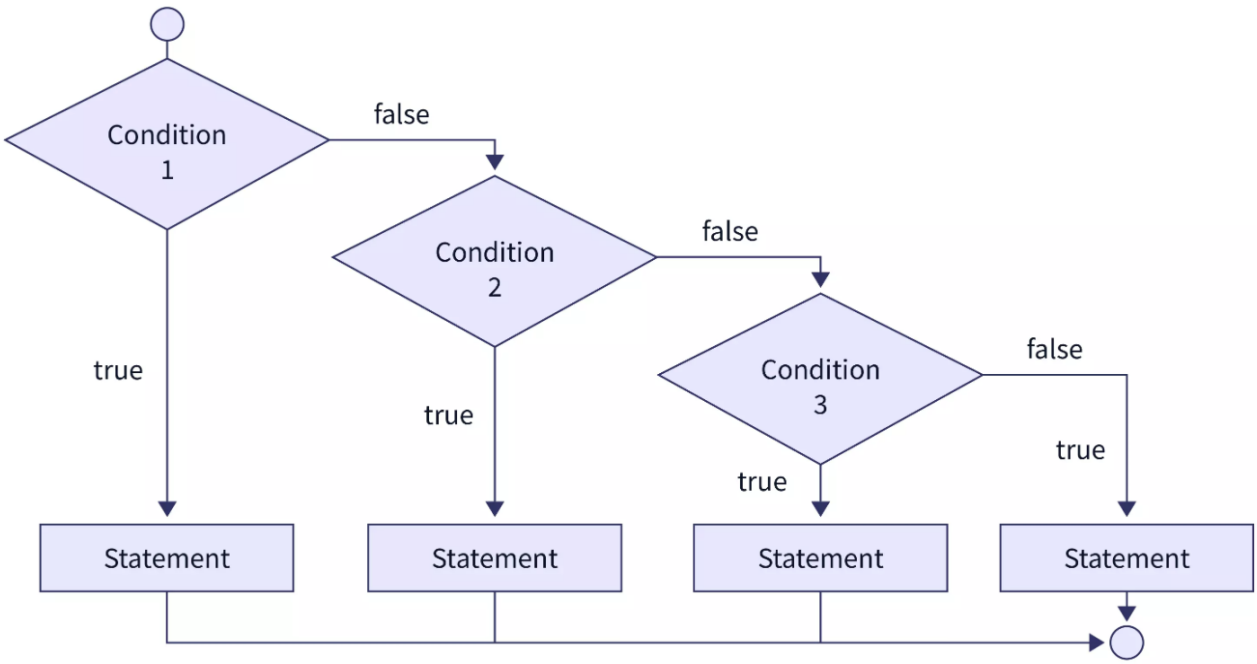
Example 5: char with if-else

```
char grade = 'B';

if (grade == 'A' || grade == 'B') {
    System.out.println("Well done!");
} else {
    System.out.println("Work harder for a better grade.");
```

```
}

```

In this example, the `if` condition checks if the `grade` variable is 'A' or 'B'. If it is, the message "Well done!" is printed. Otherwise, the `else` block is executed, and the message "Work harder for a better grade." is printed.

Example 6: if, else if & else



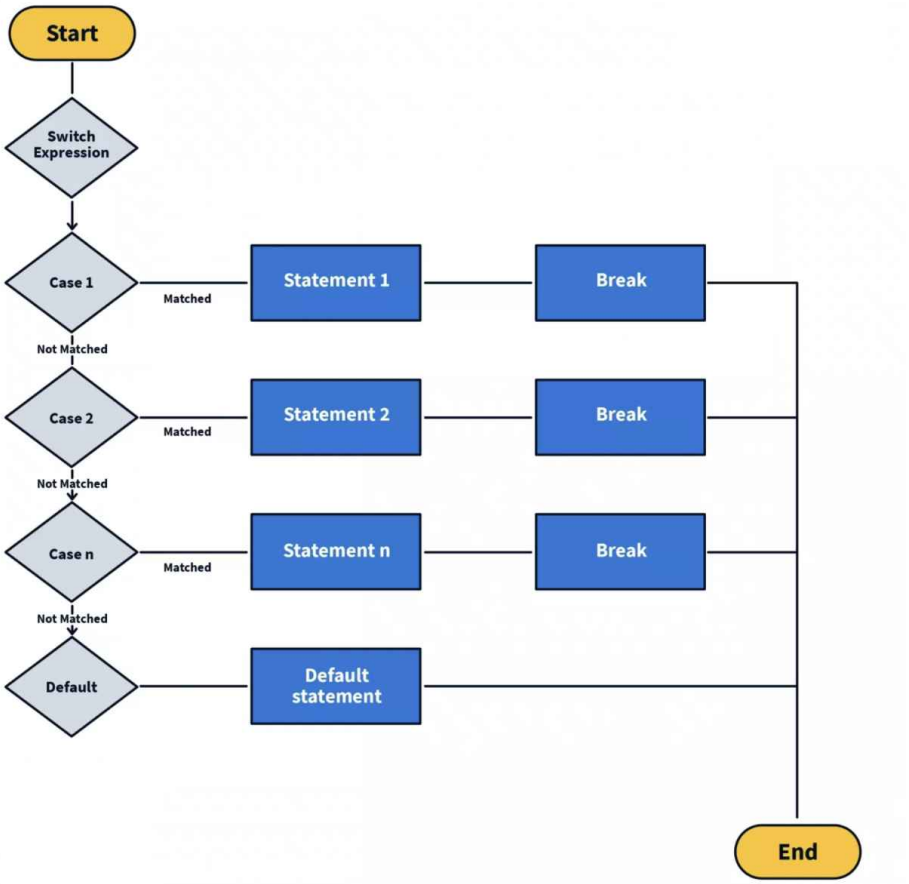

```
int score = 85;

if (score >= 90) {
    System.out.println("Excellent!");
} else if (score >= 80) {
    System.out.println("Good job!");
} else if (score >= 70) {
    System.out.println("Keep it up!");
} else {
    System.out.println("You can do better.");
}
```

In this example, the `if`, `else if`, and `else` statements are used to determine the grade based on the `score` variable. The conditions are checked sequentially, and the appropriate message is printed based on the score range.

Challenge

```java
int numericGrade = 78;
boolean isPassed = true;
char letterGrade = 'Z';

if (numericGrade <= 0) { // fulfill ?
    throw new IllegalArgumentException("Grade must be larger than zero!");
}

if (numericGrade >= 50) { // fulfill ?
    isPassed = true;
} else {
    isPassed = false;
}

// What's next?
if (numericGrade >= 90) {
    letterGrade = 'A';
} else if (numericGrade >= 80) {
    letterGrade = 'B';
} else if (numericGrade >= 70) {
    letterGrade = 'C';
} else if (numericGrade >= 60) {
    letterGrade = 'D';
} else if (numericGrade >= 50) {
    letterGrade = 'E';
} else {
    letterGrade = 'F';
}

System.out.println("isPassed: " + isPassed); // isPassed = ?
System.out.println("numericGrade: " + numericGrade); // numericGrade = ?
System.out.println("letterGrade: " + letterGrade); // letterGrade = ?
```

Switch statements



In Java, the `switch` statement provides a way to perform different actions based on the value of a variable or expression. It offers a more concise alternative to multiple `if-else if-else` statements.

Here's an introduction to the `switch` statement with examples covering various scenarios:

The basic syntax of the `switch` statement is as follows:

```
switch (expression) {
    case value1:
        // Code to be executed if expression matches value1
        break;
    case value2:
        // Code to be executed if expression matches value2
        break;
    // ...
    default:
        // Code to be executed if expression does not match any case
        break;
}
```

Here's an explanation of the different parts of the `switch` statement:

- `expression` : A variable or expression whose value is being compared.
- `case value` : Specifies the possible values that the expression can match.
- `break` : Terminates the `switch` block and prevents the execution of subsequent cases. It is typically included after each case.
- `default` : Represents the code to be executed if the expression does not match any of the cases. It is optional and can appear at any position within the `switch` block.

Now, let's look at examples that cover various scenarios:

Example 1: Matching Integer Values

```
int dayOfWeek = 3;
String dayName;

switch (dayOfWeek) {
    case 1:
        dayName = "Sunday";
        break;
    case 2:
        dayName = "Monday";
        break;
    case 3:
        dayName = "Tuesday";
        break;
    case 4:
        dayName = "Wednesday";
        break;
    case 5:
        dayName = "Thursday";
        break;
    case 6:
        dayName = "Friday";
        break;
    case 7:
        dayName = "Saturday";
        break;
    default:
        dayName = "Invalid day";
        break;
}

System.out.println("Day of the week: " + dayName);
```

In this example, the `switch` statement matches the value of `dayOfWeek` and assigns the corresponding day name to the `dayName` variable. The output will be "Day of the week: Tuesday" since `dayOfWeek` is 3.

Example 2: Matching Characters

```
char grade = 'B';

switch (grade) {
    case 'A':
    case 'B':
        System.out.println("Excellent!");
        break;
    case 'C':
        System.out.println("Good job!");
        break;
    case 'D':
        System.out.println("You passed.");
        break;
    case 'F':
        System.out.println("You failed.");
        break;
    default:
        System.out.println("Invalid grade.");
        break;
}

```

In this example, the `switch` statement matches the value of `grade` and prints a corresponding message. Since `grade` is 'B', the output will be "Excellent!".

Example 3: Matching Strings

```
String fruit = "apple";

switch (fruit) {
    case "apple":
        System.out.println("It's an apple.");
        break;
    case "banana":
        System.out.println("It's a banana.");
        break;
    case "orange":
```

```
        System.out.println("It's an orange.");
        break;
    default:
        System.out.println("Unknown fruit.");
        break;
}
```

In this example, the `switch` statement matches the value of `fruit` and prints the corresponding message. Since `fruit` is "apple", the output will be "It's an apple.".

These examples demonstrate the usage of the `switch` statement in Java. It allows you to simplify decision-making based on different values of a variable or expression, resulting in more concise and readable code.

Challenge 1

```
int numericGrade = 20;
char letterGrade = ' ';

switch (numericGrade) {
    case 90: {
        letterGrade = 'A';
    }
    case 80: {
        letterGrade = 'B';
    }
    case 70: {
        letterGrade = 'C';
    }
    case 60: {
        letterGrade = 'D';
    }
    case 50: {
        letterGrade = 'E';
    }
    default: {
        letterGrade = 'F';
    }
}

System.out.println("letterGrade: " + letterGrade); // letterGrade = ?
```

Challenge 2

```
int numericGrade = 78;

// "Case 78" fulfill? What's next?
switch (numericGrade) {
    case 50: {
        letterGrade = 'B';
    }
    case 78: {
        letterGrade = 'A';
    }
    case 80: {
        letterGrade = 'D';
    }
    default: {
        letterGrade = 'C';
    }
}

System.out.println("The letter grade is " + letterGrade); // letterGrade = ?
```

Questions

- Practice if, else if & else statement as much as possible.
- Did you try various logical operators & if-else statement together?
- Break is important & good practice when you are using a switch. Try to code switch statement yourself & make sure you get familiar with switch.