Ministry of Higher Education
Jami University
Scientific assistantship
Faculty (Computer Science)
Department of Information Systems

Conditions in java

In Java, **conditions** are typically used to control the flow of execution based on certain criteria. Conditional statements allow a program to execute different code blocks depending on the result of an evaluation. The main types of conditional statements in Java include:

1. **`if` statement**
2. **`else` statement**
3. **`else if` statement**
4. **`switch` statement**
5. **Ternary operator**

1. **`if` Statement**:

An `if` statement is used to execute a block of code if a specified condition is true.

Example 1: Basic `if` statement

```java
int x = 5;
if (x > 0) {
   System.out.println("x is positive");
}
```

Output:

```
x is positive
```

Saifullah Haidari

Example 2: `if` with boolean condition

```java
boolean isRaining = true;
if (isRaining) {
   System.out.println("Take an umbrella");
}
```

Output:

```
Take an umbrella
```

2. **`else` Statement**:

The `else` statement is used to execute a block of code when the condition in the `if` statement is false.

Example 3: Using `else`

```java
int age = 15;
if (age >= 18) {
   System.out.println("You are an adult");
} else {
   System.out.println("You are a minor");
}
```

Output:

```
You are a minor
```

Saifullah Haidari

3. **`else if` Statement**:

The `else if` statement allows for multiple conditions to be tested in sequence. Once a condition is true, its block of code is executed, and the rest are skipped.

Example 4: Using `else if`

```java
int num = 0;
if (num > 0) {
   System.out.println("Positive");
} else if (num < 0) {
   System.out.println("Negative");
} else {
   System.out.println("Zero");
}
```

Output:

```
Zero
```

Example 5: Multiple `else if`

```java
int score = 85;
if (score >= 90) {
   System.out.println("A");
} else if (score >= 80) {
   System.out.println("B");
} else if (score >= 70) {
```

```
    System.out.println("C");
} else {
    System.out.println("F");
}
```

Output:

```
B
```

4. **`switch` Statement**:

The `switch` statement is used to execute one block of code among many based on the value of a variable.

Example 6: Basic `switch`

```java
int day = 3;
switch (day) {
    case 1: System.out.println("Monday"); break;
    case 2: System.out.println("Tuesday"); break;
    case 3: System.out.println("Wednesday"); break;
    default: System.out.println("Invalid day");
}
```

Output:

```
Wednesday
```

Example 7: `switch` with `char`

```java
char grade = 'A';
switch (grade) {
   case 'A': System.out.println("Excellent"); break;
   case 'B': System.out.println("Good"); break;
   case 'C': System.out.println("Fair"); break;
   default: System.out.println("Fail");
}
```

Output:

```
Excellent
```

Example 8: `switch` without `break` (fall-through behavior)

```java
int month = 2;
switch (month) {
   case 1:
   case 2:
   case 3:
      System.out.println("First Quarter");
      break;
   default:
      System.out.println("Other Quarter");
}
```

Output:

Saifullah Haidari

```

First Quarter

```

5. **Ternary Operator**:

The ternary operator is a shorthand for `if-else` statements, using the format `condition ? value_if_true : value_if_false`.

Example 9: Ternary operator for a condition

```java

int a = 10, b = 20;

int max = (a > b) ? a : b;

System.out.println("Max value: " + max);

```

Output:

```

Max value: 20

```

6. **Nested `if` Statements**:

Conditions can also be **nested** inside other `if` or `else` statements.

Example 10: Nested `if`

```java

int x = 10;

if (x > 0) {

    if (x % 2 == 0) {

        System.out.println("x is positive and even");

    } else {
```

```
        System.out.println("x is positive and odd");
    }
}
```

Output:

```
x is positive and even
```

7. **Complex Conditions**:

You can combine conditions using logical operators like `&&` (AND) and `||` (OR).

Example 11: Using `&&` (AND) operator

```java
int x = 10, y = 20;
if (x > 5 && y < 30) {
    System.out.println("Both conditions are true");
}
```

Output:

```
Both conditions are true
```

Example 12: Using `||` (OR) operator

```java
int x = 5, y = 50;
if (x < 10 || y > 40) {
    System.out.println("At least one condition is true");
```

}
```
Output:
```
At least one condition is true
```

### 8. **`switch` with Strings**:

Java allows `switch` statements with `String` values (since Java 7).

#### Example 13: `switch` with `String`

```java
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;
    default: System.out.println("Unknown fruit");
}
```
Output:
```
It's an Apple
```

9. **Checking `null` in Conditions**:

You can check for `null` values in conditions to avoid `NullPointerException`.

Example 14: `null` check

```java
Saifullah Haidari

```
String name = null;
if (name == null) {
    System.out.println("Name is null");
} else {
    System.out.println("Name is " + name);
}
```

Output:

```
Name is null
```

10. **Condition with Arrays**:

You can use conditions to check array contents.

Example 15: Check if array contains an element

```java
int[] numbers = {1, 2, 3, 4, 5};
int searchNumber = 3;
boolean found = false;
for (int number : numbers) {
    if (number == searchNumber) {
        found = true;
        break;
    }
}
if (found) {
    System.out.println("Number found");
} else {
```

```
    System.out.println("Number not found");
}
```

Output:

```
Number found
```

Summary:

- **`if` statements** are used for single conditions.
- **`else if` and `else`** allow for multiple conditions.
- **`switch` statements** are useful for multiple possible values of a variable.
- The **ternary operator** is a shorthand for `if-else`.
- Combining logical operators like `&&` and `||` allows complex conditional expressions.