`switch` statement:

```java
switch (expression) {
    case value1:
        // Code to execute if expression matches value1
        break;
    case value2:
        // Code to execute if expression matches value2
        break;
    // ...
    default:
        // Code to execute if no cases match
}
```

Here's how the `switch` statement works in more detail:

1. **Basic `switch` Statement**:
   The `switch` statement evaluates the expression within the parentheses and then compares its value against the different `case` labels. If a match is found, the corresponding block of code is executed until a `break` statement is encountered.

   ```java
   int dayOfWeek = 3;

   switch (dayOfWeek) {
       case 1:
           System.out.println("Sunday");
           break;
       case 2:
           System.out.println("Monday");
           break;
       case 3:
           System.out.println("Tuesday");
           break;
       // ... (other cases)
       default:
           System.out.println("Invalid day");
   }
   ```

2. **Fall-Through Behavior**:
   In Java, unlike some other programming languages, there's no automatic "fall-through" behavior in `switch` statements. Once a matching `case` is found and executed, the execution stops unless a `break` statement is encountered.

   ```java
   int num = 2;

   switch (num) {
       case 1:
           System.out.println("One");
           // No 'break' here, so execution continues to the next case
       case 2:
           System.out.println("Two");
           break;
       default:
           System.out.println("Other");
   }
   // Output: Two
   ```

3. **`default` Case**:
   The `default` case is executed if none of the `case` values match the expression value. It's not required, but it's a good practice to include it for handling unexpected or unknown cases.

4. **`switch` with Other Data Types**:
   The `switch` statement supports `byte`, `short`, `char`, `int`, `enum`, and `String` data types. It doesn't support floating-point types like `float` or `double`.

   ```java
   String month = "June";

   switch (month) {
       case "January":
       case "February":
       case "March":
           System.out.println("Winter season");
           break;
       case "June":
       case "July":
       case "August":
           System.out.println("Summer season");
           break;
       // ... (other cases)
   ```

```
      default:
          System.out.println("Other season");
  }
```

The `switch` statement can be a concise and readable way to handle multiple conditions based on a single expression. However, remember that `switch` is most effective when dealing with a small number of cases and discrete values. If you have complex conditions or a large number of possible values, using `if...else if...else` might be more suitable.