

SWITCH STATEMENT IN JAVA

1. What is a switch statement and how does it work?

Answer:

A switch statement is a control flow statement that executes different code blocks based on the value of a variable. It works by comparing the variable to different case values and executes the corresponding code block. If no match is found, an optional default block is executed.

2. What data types can be used in the switch expression?

Answer:

In Java, you can use primitives (int, char, byte, short, boolean, long), their wrapper classes (Integer, Character, etc.), enums, and Strings in the switch expression.

3. What happens if no case matches the expression value?

Answer:

If no case value matches the expression, the execution jumps to the default block, if present. If no default block is provided, the switch statement does nothing.

4. Can you explain fall-through in a switch statement?

Answer:

By default, if a break statement is not present at the end of a case block, execution falls through to the next case block, even if it doesn't match the expression value. This can be useful in some cases but can also lead to unexpected behavior if not used carefully.

5. When should you use a switch statement instead of if-

else statements?

Answer:

Switch statements are generally considered more readable and maintainable than long chains of if-else statements when dealing with multiple comparisons against the same variable. However, if the logic gets complex or the variable types aren't compatible with switch, if-else might be a better choice.

6. How can you improve the readability and maintainability of a switch statement?

Answer:

Use clear and concise case values.

Use meaningful variable names.

Include comments to explain complex logic.

Consider using fall-through with caution and explicitly document its usage.

7. What are the limitations of switch statements in Java?

Answer:

Java switch statements can't handle ranges or floating-point numbers directly. This can be overcome by using workarounds but might affect readability.

8. How does Java 14 enhance switch statements?

Answer:

Java 14 introduced pattern matching in switch expressions, allowing for more flexible comparisons and improved readability.