three types of comments:

code. They help developers understand the code better but are ignored by the Java compiler. Java supports three types of comments:

1. **Single-line comments**

2. **Multi-line comments**

3. **Documentation comments (Javadoc)**

### 1. **Single-line Comments**:

These comments begin with `//`. Everything after `//` on that line is considered a comment.

#### Example:

```java

// This is a single-line comment

int age = 25;  // Declaring an integer variable

```

### 2. **Multi-line Comments**:

Multi-line comments begin with `/*` and end with `*/`. These comments can span multiple lines and are useful for large blocks of text.

#### Example:

```java

/*

  This is a multi-line comment.

  It can span multiple lines.
```

```
*/
int sum = 0;  /* This is an inline multi-line comment */
```

### 3. **Documentation Comments (Javadoc)**:

Javadoc comments are used to generate HTML documentation for your code. These comments begin with `/**` and end with `*/`. They are commonly placed above classes, methods, or fields to describe their purpose.

#### Example:

```java
/**
 * This class represents a simple calculator.
 * It performs basic arithmetic operations.
 */
public class Calculator {

    /**
     * This method adds two numbers.
     * @param a The first number
     * @param b The second number
     * @return The sum of a and b
     */
    public int add(int a, int b) {
        return a + b;
    }
}
```

### Examples of Comments in Different Scenarios:

1. **Using a single-line comment to explain a variable**:
   ```java
   int age = 30;  // Age of the person
   ```

2. **Using a multi-line comment to explain a method**:
   ```java
   /*
    * This method calculates the area of a rectangle.
    * It takes two parameters: width and height.
    */
   public int calculateArea(int width, int height) {
       return width * height;
   }
   ```

3. **Adding comments within loops**:
   ```java
   for (int i = 0; i < 10; i++) {
       // Printing the value of i
       System.out.println("i = " + i);
   }
   ```

4. **Documenting a class using Javadoc comments**:
   ```java
   /**
   ```

```
 * The Person class stores the name and age of a person.
 */
public class Person {
    String name;
    int age;

    /**
     * Constructor to initialize a new person.
     * @param name The person's name
     * @param age The person's age
     */
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
```

5. **Commenting out a block of code temporarily**:

```java
/*
int result = add(5, 10);
System.out.println(result);
*/
```

### Key Points:

- **Single-line comments (`//`)** are for brief explanations.

- **Multi-line comments (`/* */`)** are for larger blocks of explanations or for temporarily disabling code.

- **Javadoc comments (`/** */`)** are for generating HTML documentation.

Using comments effectively helps to maintain and understand code more easily, especially in larger projects.