Calling Java from PL/SQL is one of Oracle's most elegant integration features. You load Java into the database, publish it, and then call it like any PL/SQL function.

Here's a clean, minimal example that shows the whole flow without unnecessary complexity.

1. Write a simple Java method

This Java code will live *inside the Oracle database*.

```
public class HelloJava {
    public static String sayHello(String name) {
        return "Hello, " + name;
    }
}
```

2. Load the Java code into Oracle

Use `loadjava` or SQL Developer, or simply embed it in SQL:

```
CREATE OR REPLACE AND RESOLVE JAVA SOURCE NAMED "HelloJava" AS
public class HelloJava {
    public static String sayHello(String name) {
        return "Hello, " + name;
    }
};
/
```

Oracle compiles this Java class inside the database.

3. Publish the Java method to PL/SQL

You create a PL/SQL wrapper that maps to the Java method:

```
CREATE OR REPLACE FUNCTION say_hello(name VARCHAR2)
RETURN VARCHAR2
AS LANGUAGE JAVA
NAME 'HelloJava.sayHello(java.lang.String) return java.lang.String';
/
```

This tells Oracle:

- The PL/SQL function `say_hello` maps to `HelloJava.sayHello(String)` in Java.

4. Call the Java method from PL/SQL

```
DECLARE
    msg VARCHAR2(100);
BEGIN
    msg := say_hello('World');
    DBMS_OUTPUT.PUT_LINE(msg);
END;
/
```

Output:

Code

```
Hello, World
```