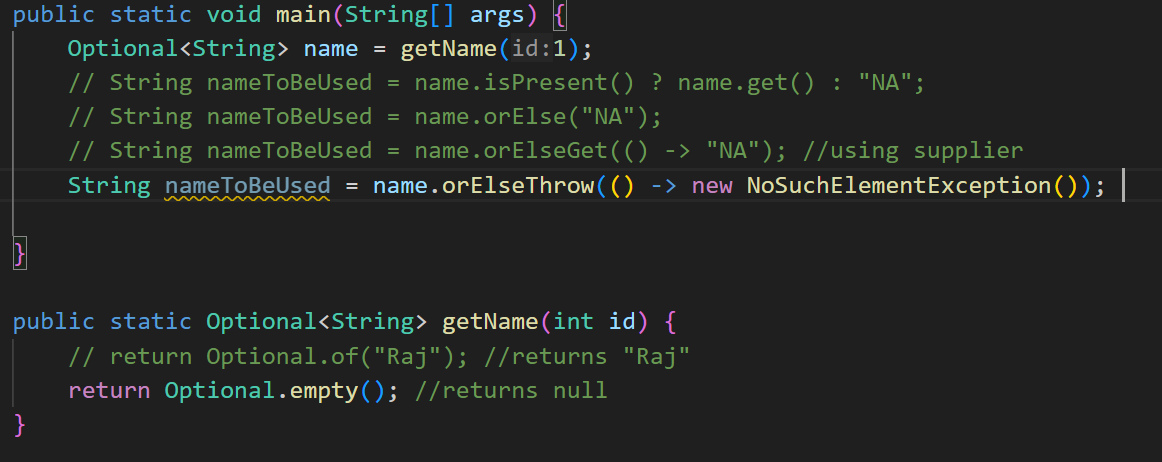
A computer screen shot of code

AI-generated content may be incorrect.



**Important things to remember** about Optional in Java:

**1. Purpose of Optional:**

* **Optional is a container object** used to represent a value that might or might not be present. It is primarily used to avoid NullPointerException and to indicate that a value may be missing, without relying on null.
* It is part of the java.util package and was introduced in **Java 8**.

**2. Avoid null References:**

* **Optional is not a replacement for null** but rather a better alternative to represent optional values without causing NullPointerExceptions.
* Using Optional signals that a value might be absent and gives a more **intentional approach** to handle missing values.

3. **Common Methods in Optional:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

4. **Creating an Optional:**

A screenshot of a computer program

AI-generated content may be incorrect.

Optional<String> Name = Optional.isNullable(name); 🡪 Creates an Optional that allows null value if name is null then it will return an empty Optional

Optional<String> Name = Optional.if(name); 🡪 Creates annon-null valued Optional

Name.ifPresent() 🡪 Returns T/F

Name.ifPresent(Consume) 🡪 Consumer 🡪 x -> sout(x) or System.out::println

Optiona.empty() returns empty optional not null or empty string

Name.get() 🡪 Returns the value or if the optional is empty then it will throw NoSuchElementException()

Name.orElse(“Default Value”)

Name.orElseGet(Supplier) 🡪 Supplier 🡪 () 🡪 “Default Name”

Name.orElseThrow(() -> new NoSuchElementException())