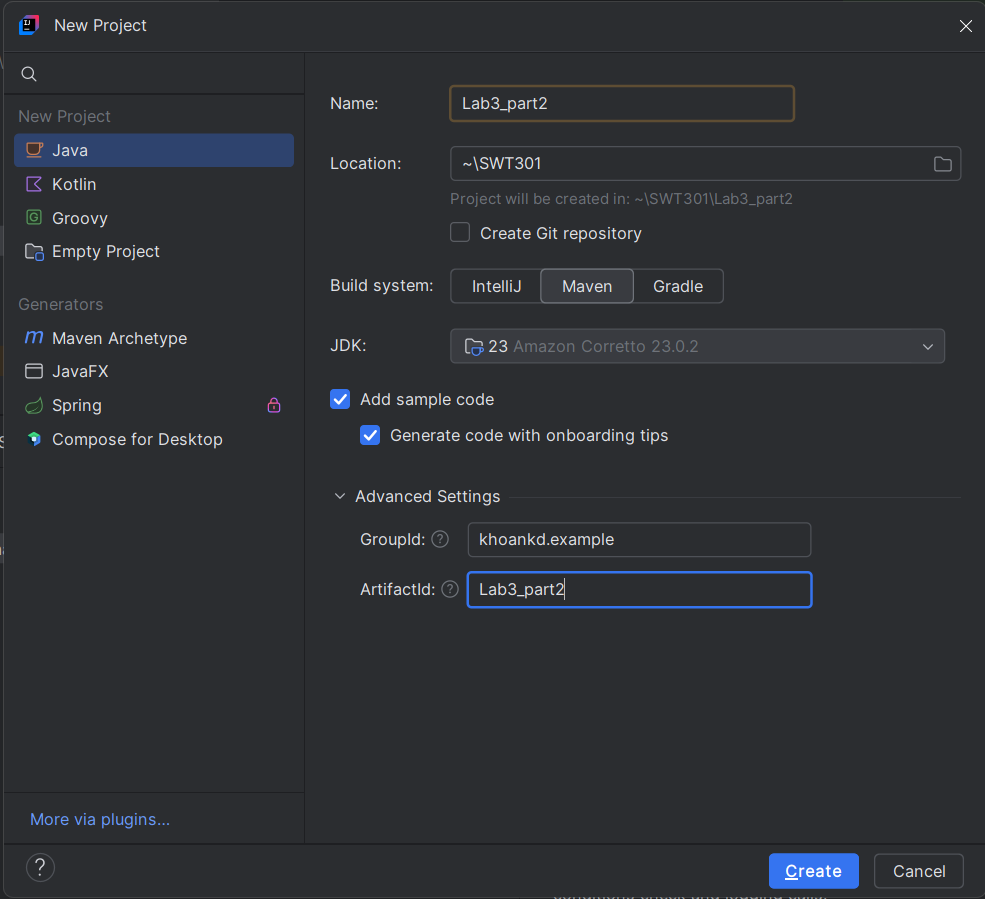
**MSSV: DE180292**

**Họ và Tên: Nguyễn Kim Đăng Khoa**

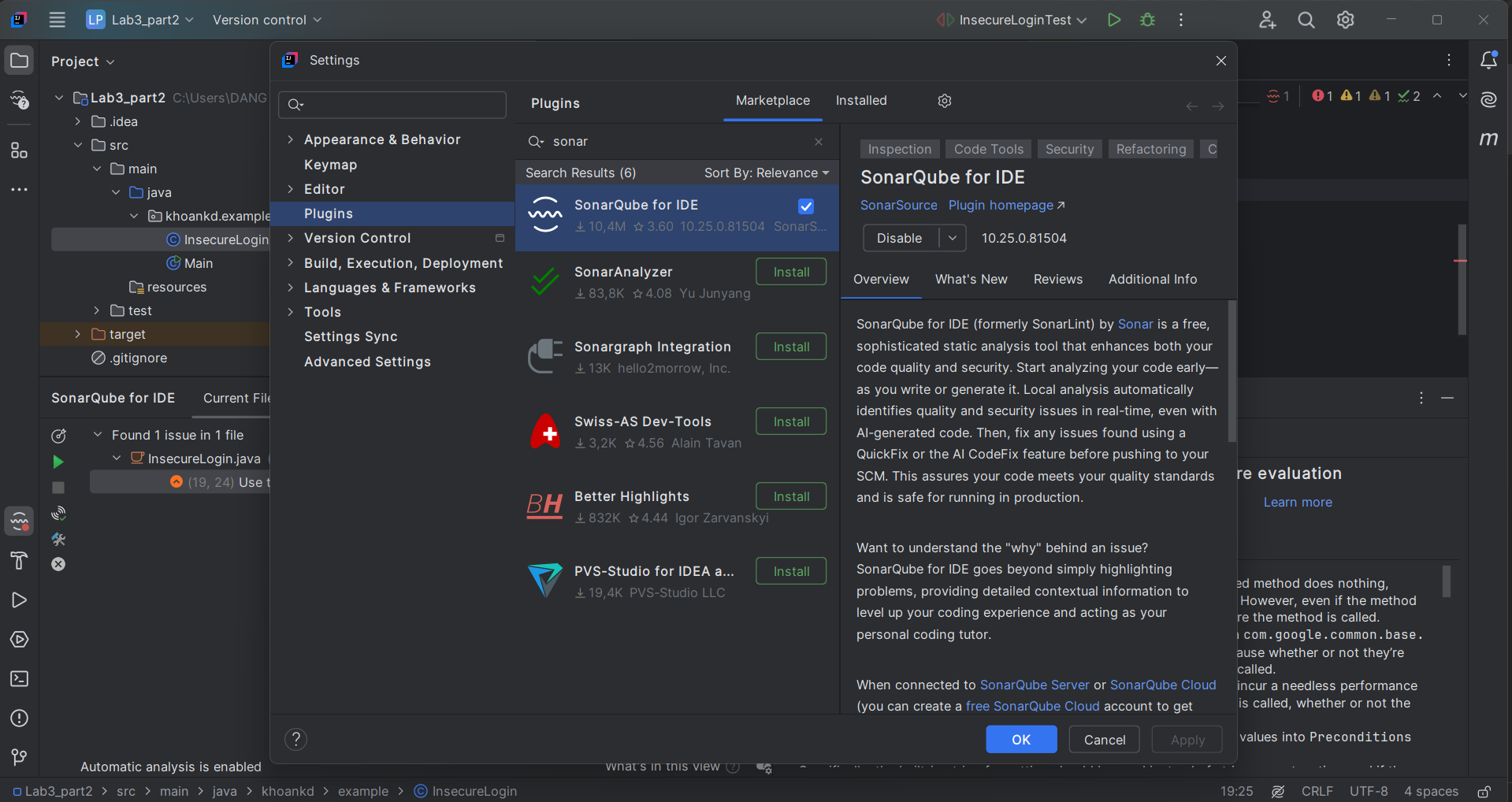
**Lớp: SE18D12**

**Phần 1: Làm theo hướng dẫn**

Bước 1: Tạo 1 project

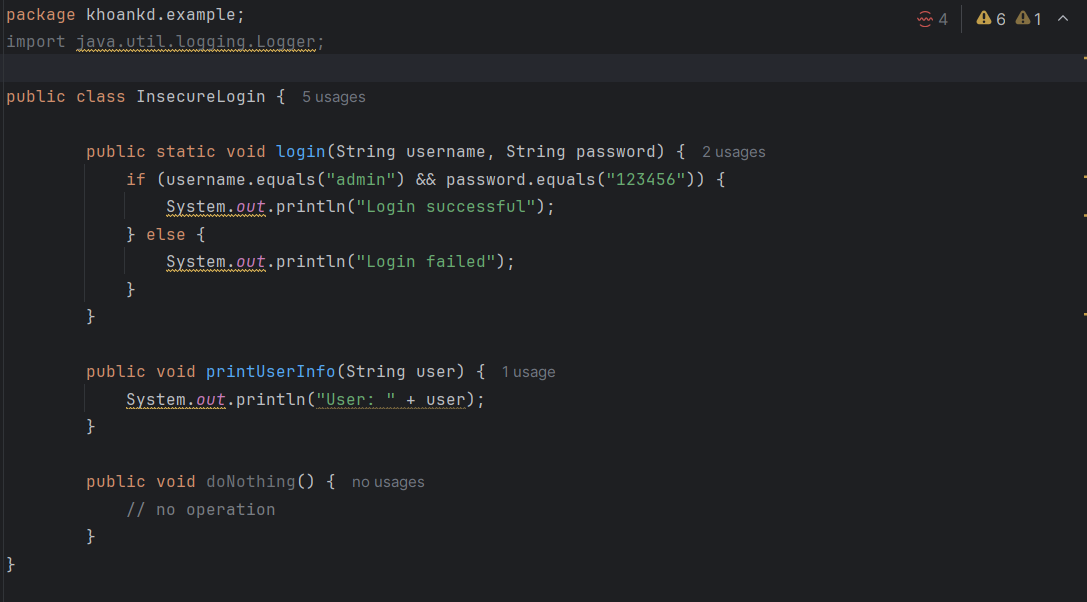
****

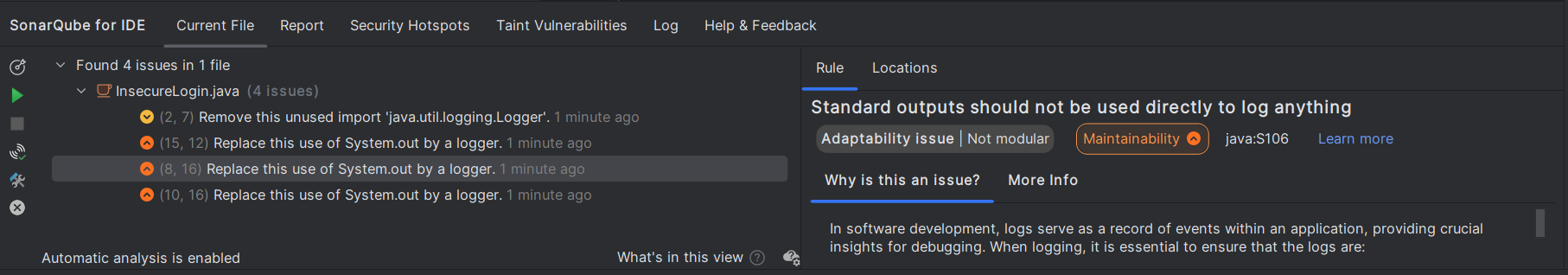
Bước 2: Tìm kiếm SonarQube for IDE và download



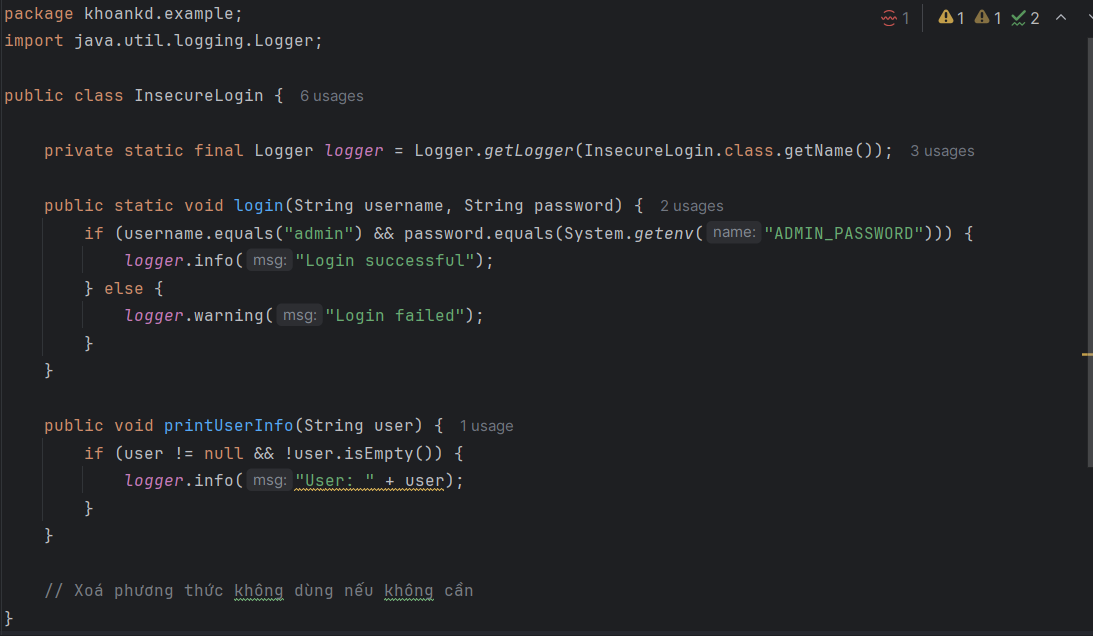
Bước 3 restart và vào lại IDE Inteliij như bình thường

Bước 4 Tạo InsecureLogin.java và Quan sát logo của Plugin mới cài bên góc phải phía trên ta thấy có 4 lỗi

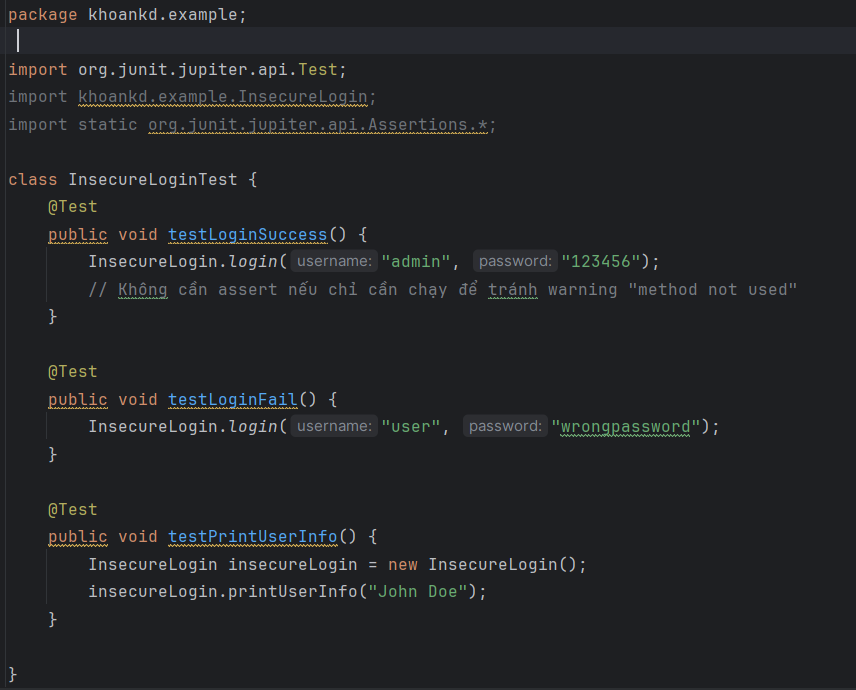




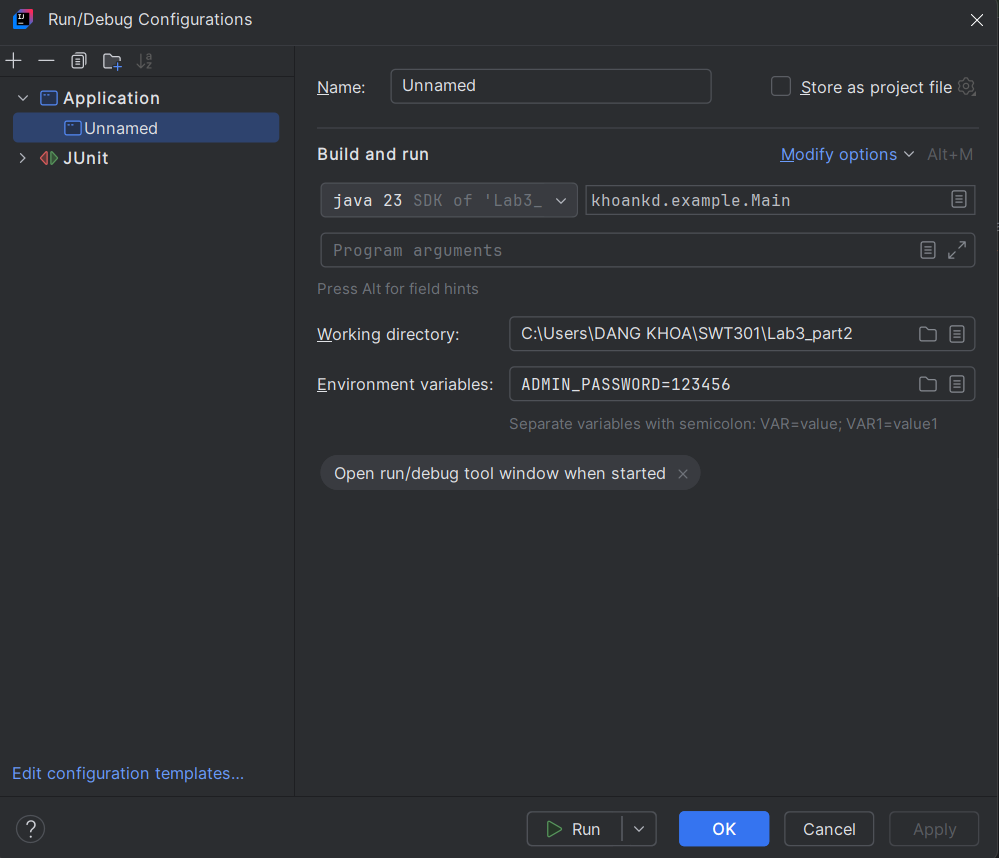
Bước 5 sửa lỗi bằng cách thay System.out bằng Logger



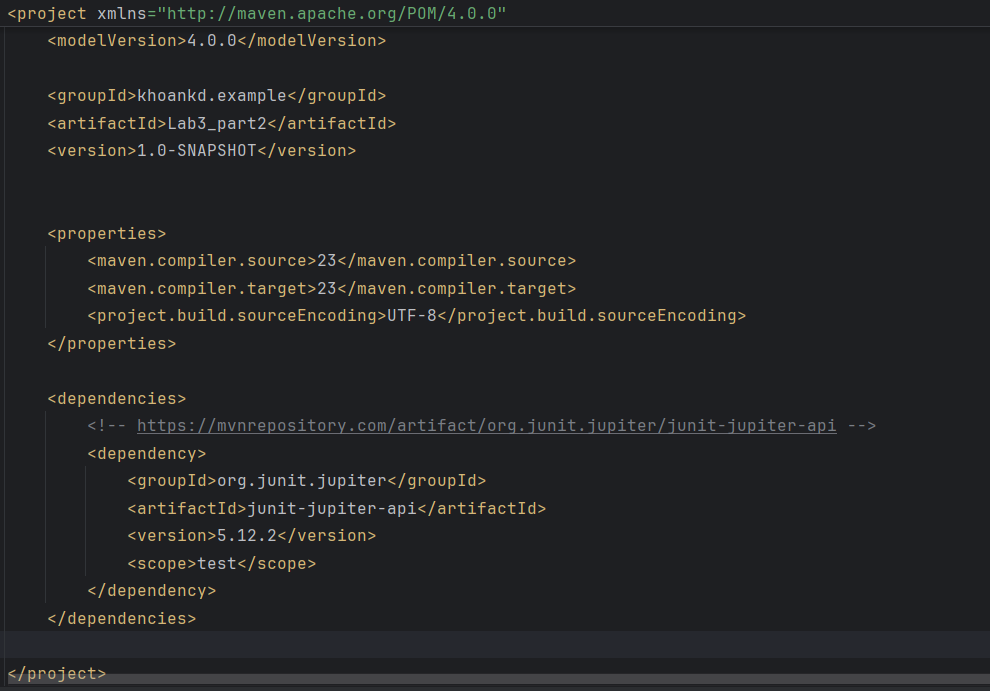
Bước 6 tạo lớp Test



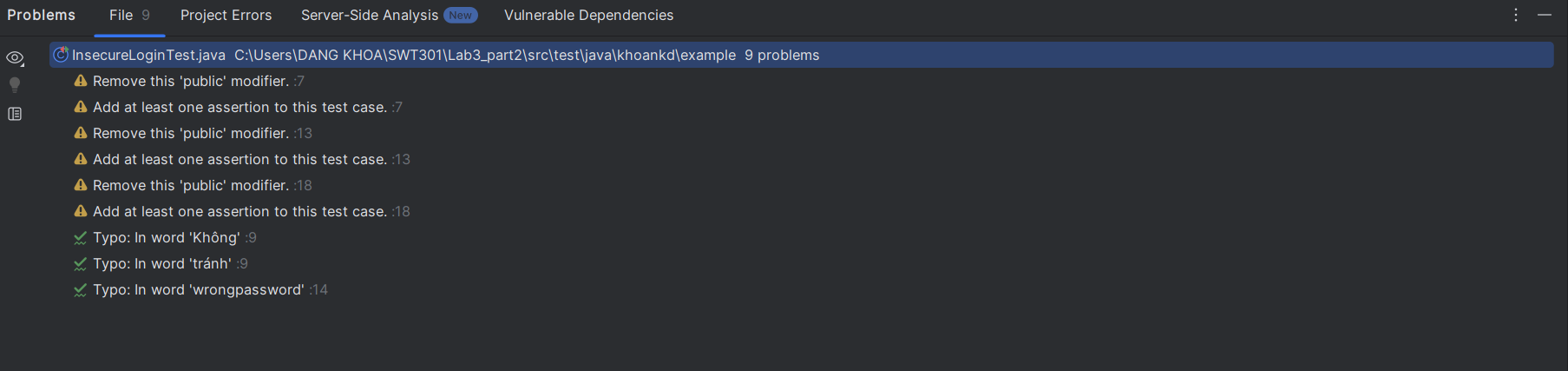
Bước 7 tạo biến môi trường



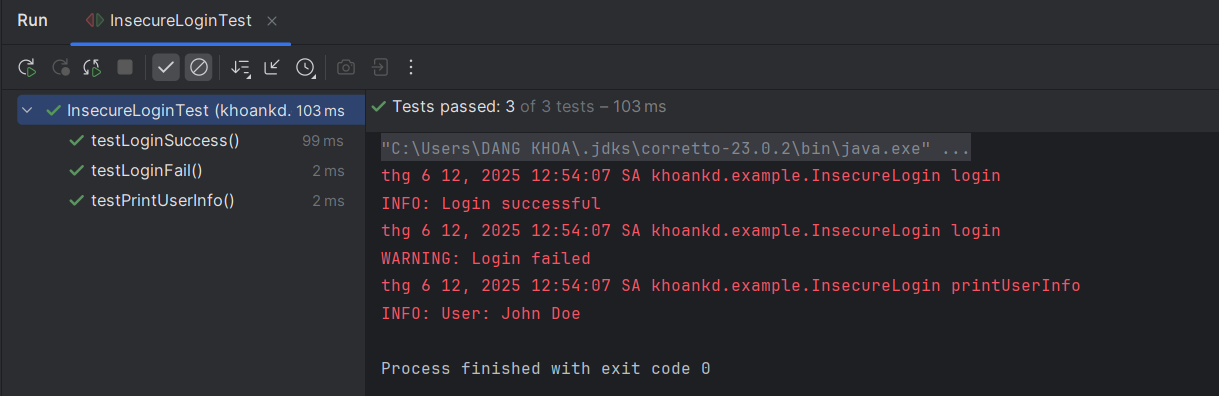
Bước 8 Cập nhật file pom.xml



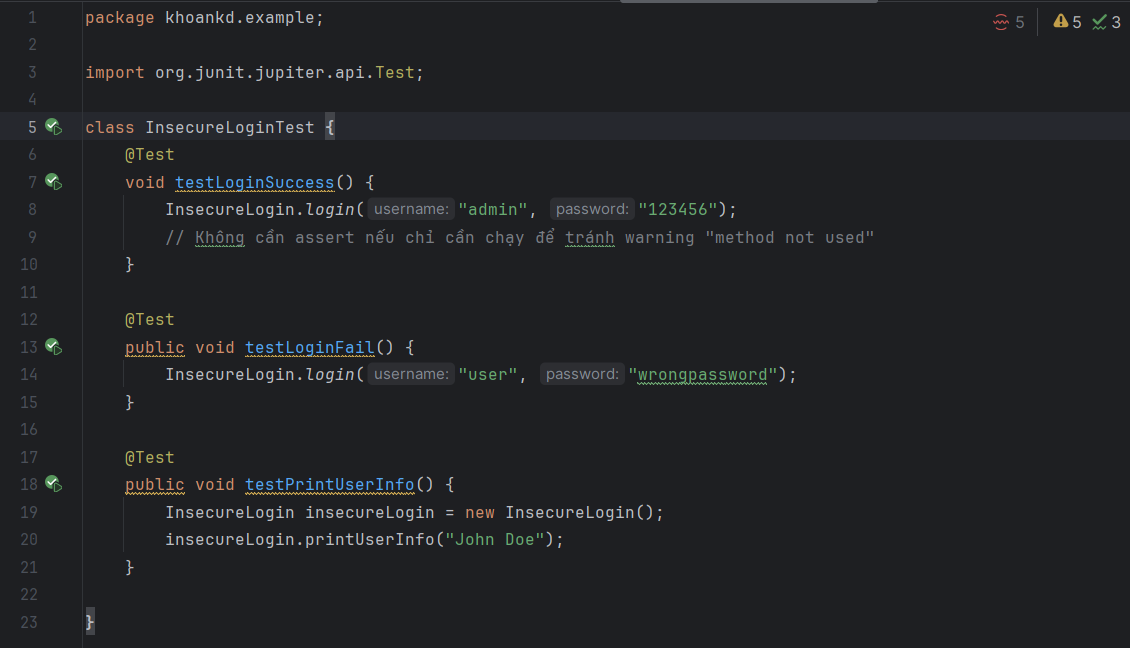
Bước 9 chạy test sẽ xuất hiện các lỗi và bấm show quick-fix để fix

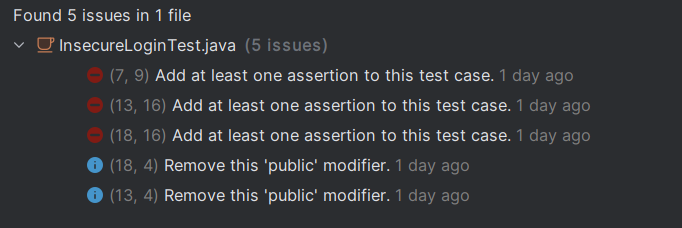


Bước 10 kết quả test sau khi cài biến môi trường



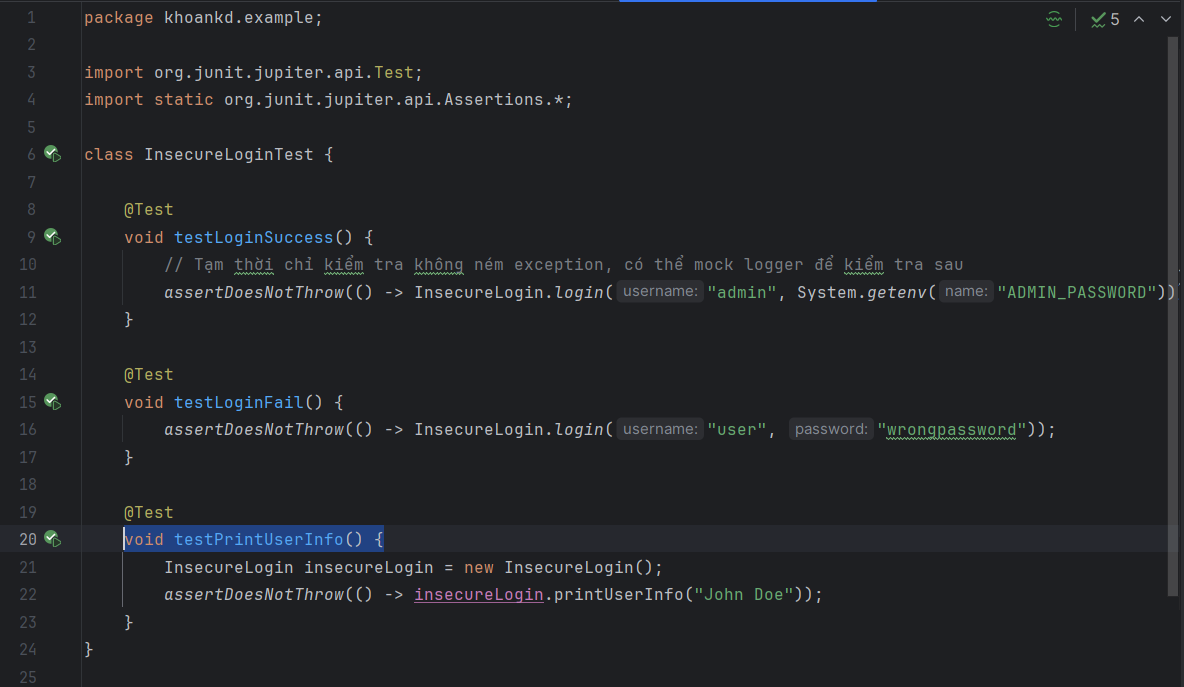
**Các lỗi của InsecureLogin.java**

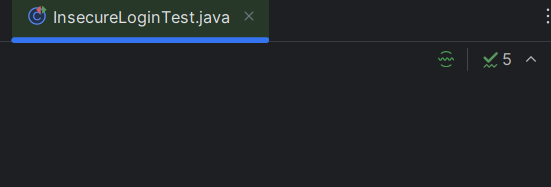
****

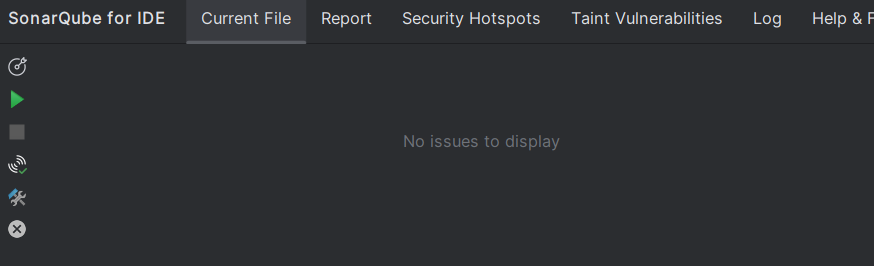
****

|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| void testLoginSuccess() {  (dòng 7) | Code smell | SonarLint cảnh báo vì các test này không đảm bảo hành vi đúng của hàm cần test. | Thêm  import static org.junit.jupiter.api.Assertions.\*; |
| public void testLoginFail() {  (dòng 13) | Code smell | SonarLint cảnh báo vì các test này không đảm bảo hành vi đúng của hàm cần test. | Thêm  import static org.junit.jupiter.api.Assertions.\*; |
| public void testPrintUserInfo() {  (dòng 18) | Code smell | SonarLint cảnh báo vì các test này không đảm bảo hành vi đúng của hàm cần test. | Thêm  import static org.junit.jupiter.api.Assertions.\*; |
| Dòng 7 | Code Convention | Trong lớp không public, các phương thức test không cần public. | void testLoginSuccess() { |
| Dòng 13 | Code Convention | Trong lớp không public, các phương thức test không cần public. | void testLoginFail() { |
| Dòng 18 | Code Convention | Trong lớp không public, các phương thức test không cần public. | void testPrintUserInfo() { |

**Kết quả sau khi sửa :**

****

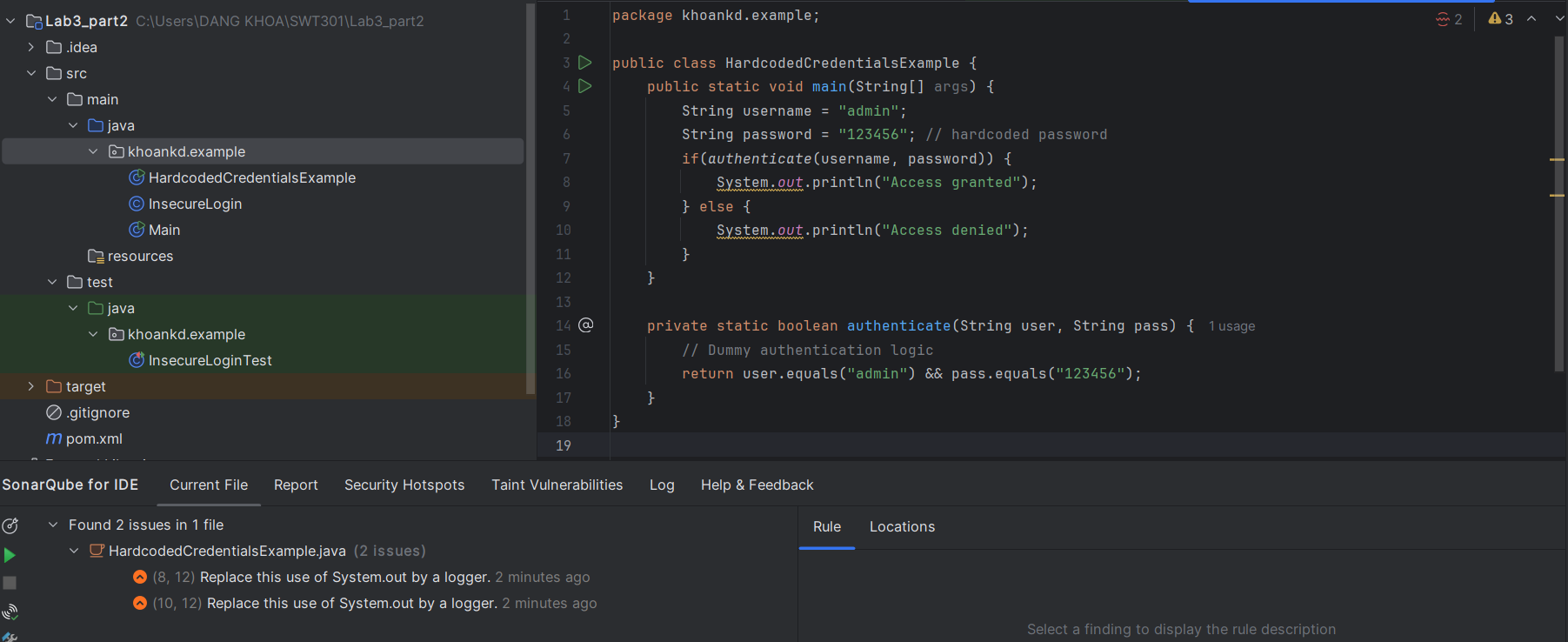


****

**Phần 2: Sửa lỗi chương trình**

1. **HardcodedCredentialsExample**

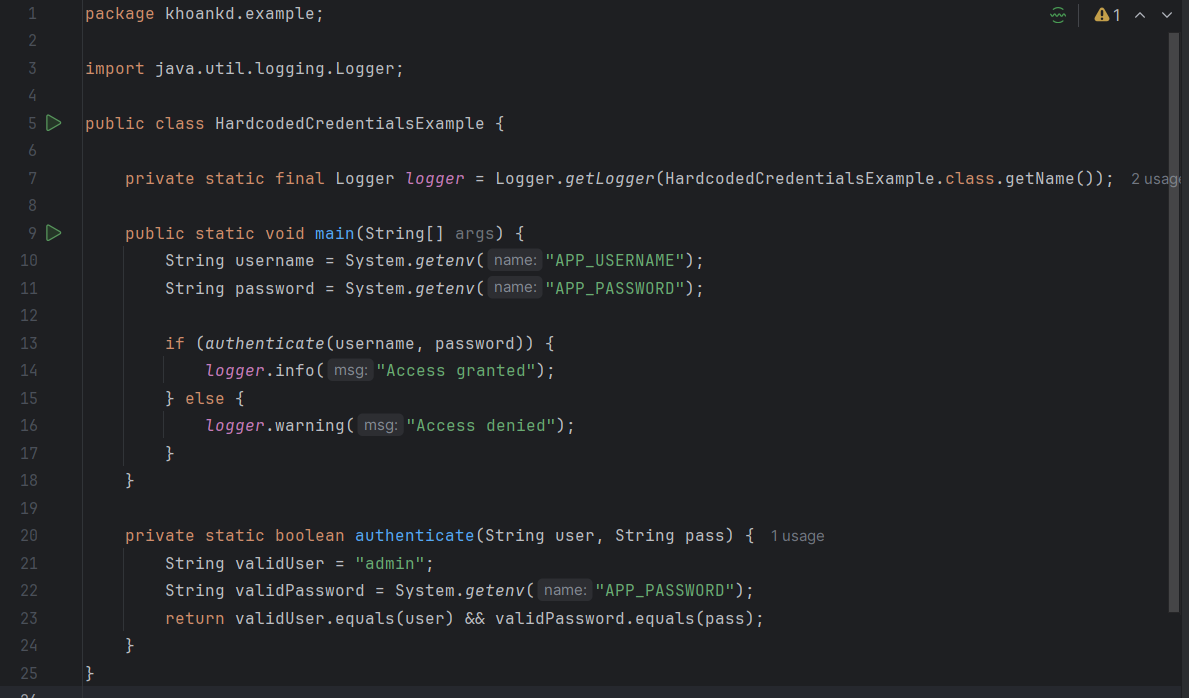
* Code nguồn ban đầu



* Các lỗi phát hiện được

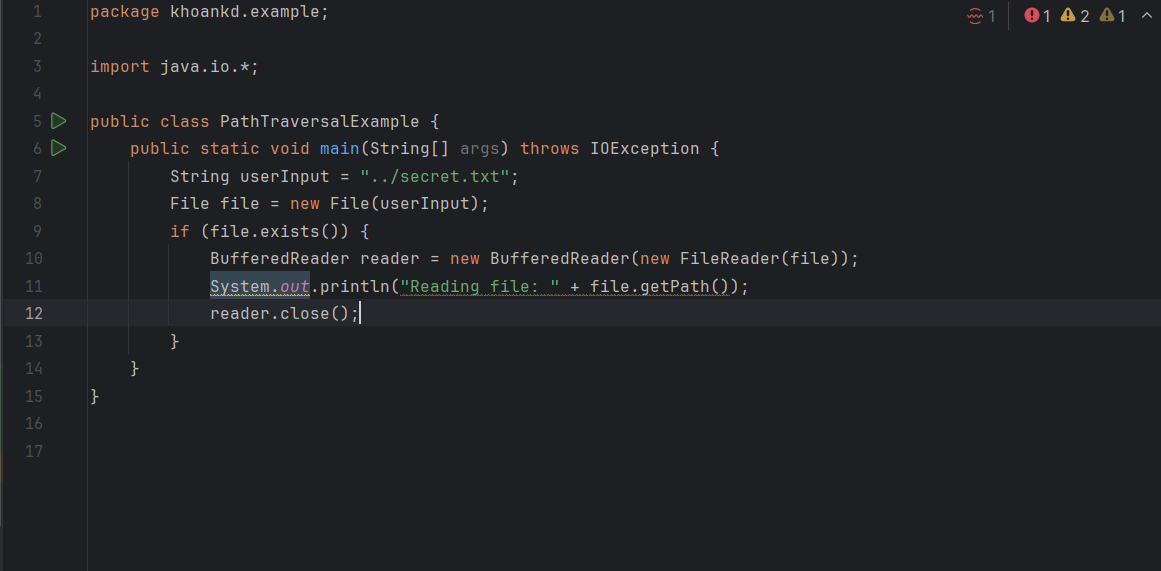
|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| Dòng 8 | |  | | --- | |  |  |  | | --- | | Code Smell | | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế |
| Dòng 10 | |  | | --- | |  |  |  | | --- | | Code Smell | | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế |

* Code sạch đã hết lỗi



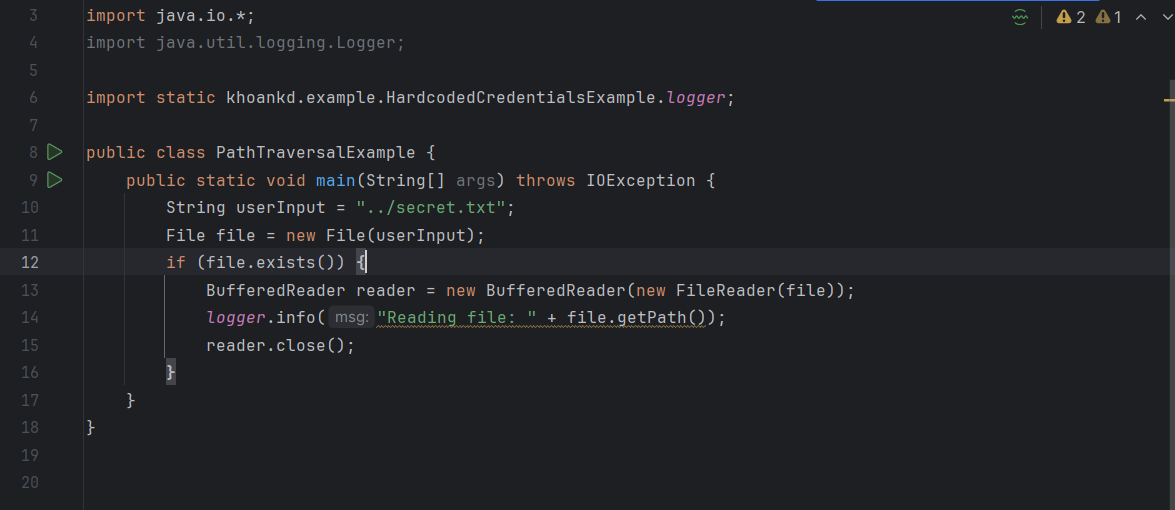
1. **PathTraversalExample**

* Code ban đầu



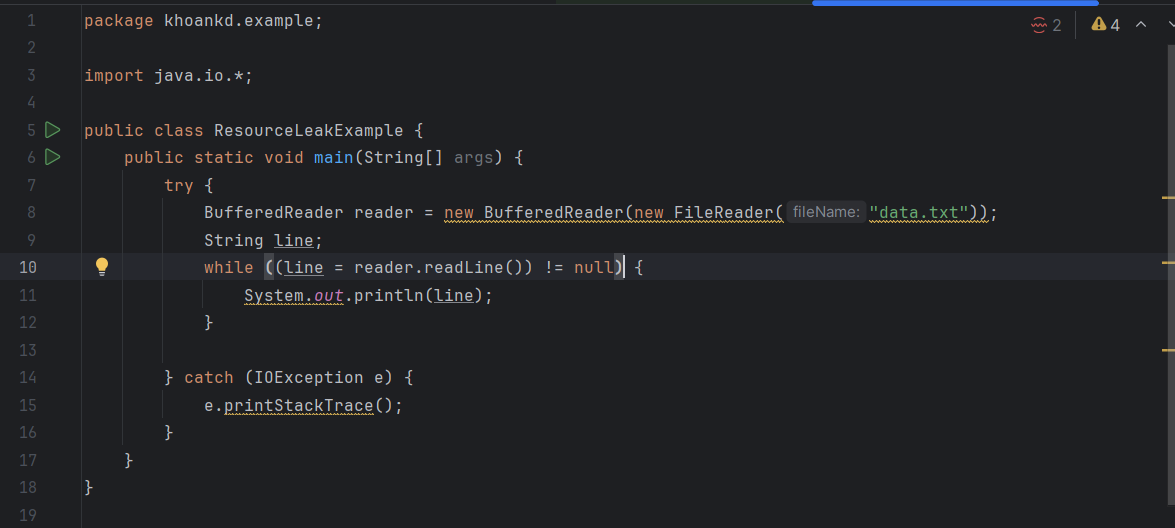
|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| Dòng 11 | |  | | --- | |  |  |  | | --- | | Code Smell | | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế |

* Code sạch hết lỗi



1. **ResourceLeakExample**

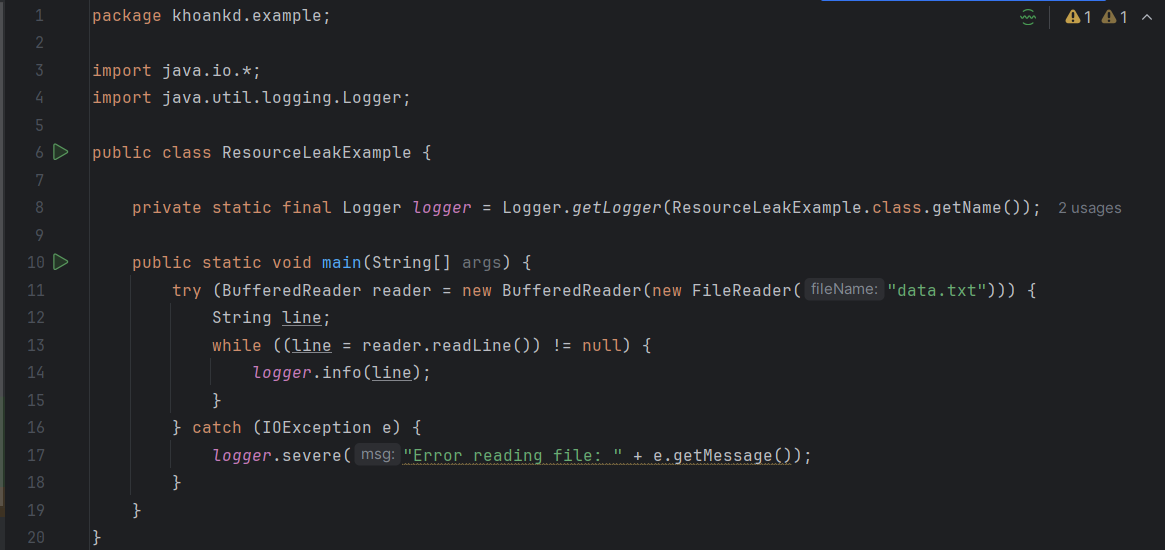
* Code ban đầu:



* Lỗi

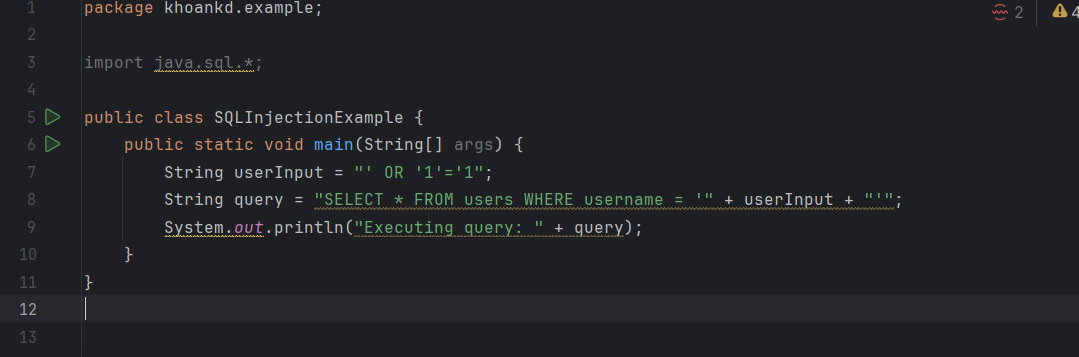
|  |  |  |  |
| --- | --- | --- | --- |
| * Vị trí | * Lỗi | * Giải thích | * Cách sửa |
| * Dòng 8 | Bug Resource Leak | Use try-with-resources or close this BufferedReader in a finally clause | * Dùng Logger thay thế |
| Dòng 11 | Code Smell | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Replace System.out with a logger |

* Code sạch và không lỗi



1. **SQLInjectionExample**

* Code ban đầu



* Lỗi

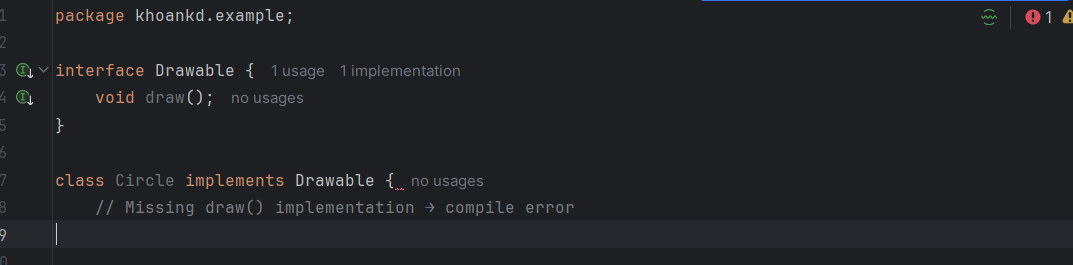
|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| Dòng 3 | Code Smell | Remove this unused import java.sql | Loại bỏ import thừa |
| Dòng 9 | Code Smell | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Replace System.out with a logger |
|  |  |  |  |

* Code sạch sau khi sửa



1. **UnimplementedInterfaceExample**

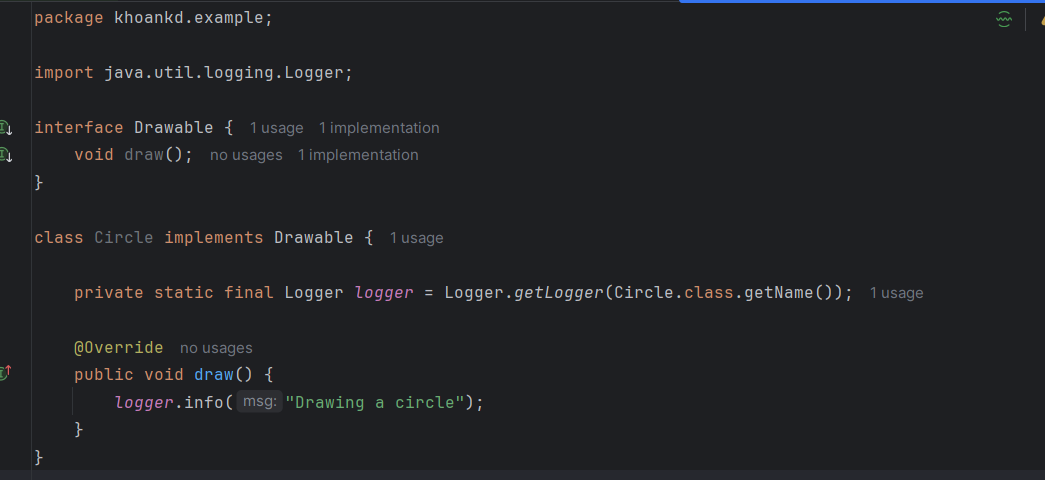
* Code ban đầu



-

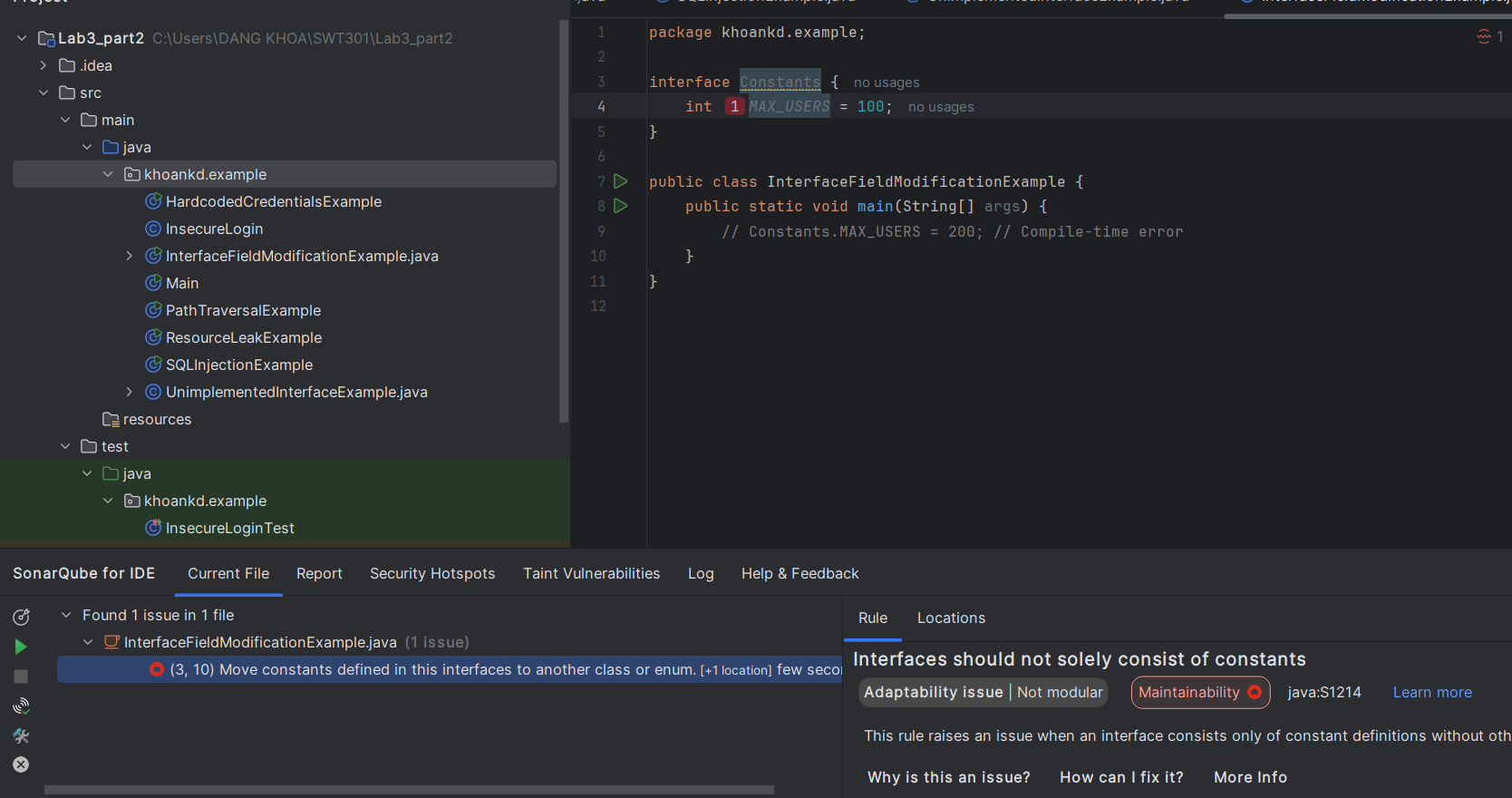
| **Vị trí** | **Lỗi** | **Giải thích** | **Cách sửa** |
| --- | --- | --- | --- |
| class Circle | Không override phương thức draw() | Lớp Circle implement interface Drawable, nhưng không cài đặt phương thức draw() → gây lỗi biên dịch do vi phạm hợp đồng interface. | Thêm phương thức public void draw() trong lớp Circle |

* Code sạch và hết lỗi



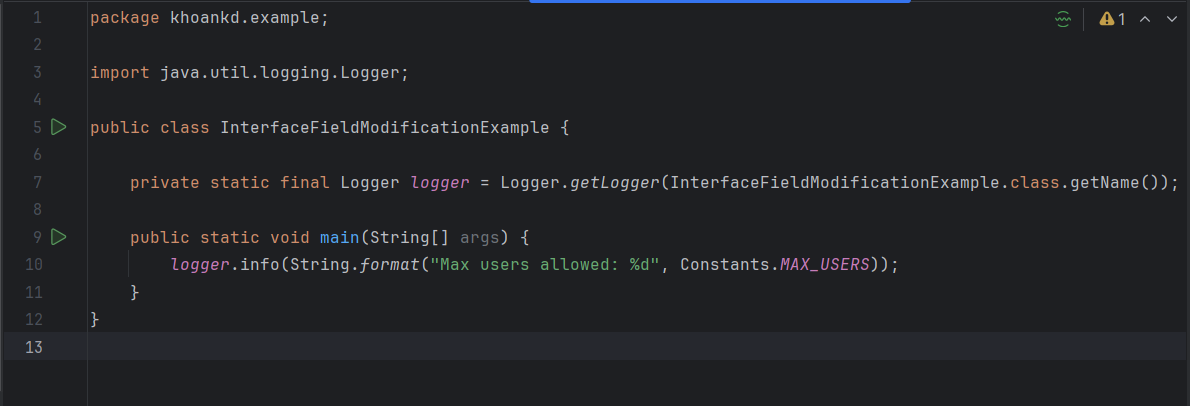
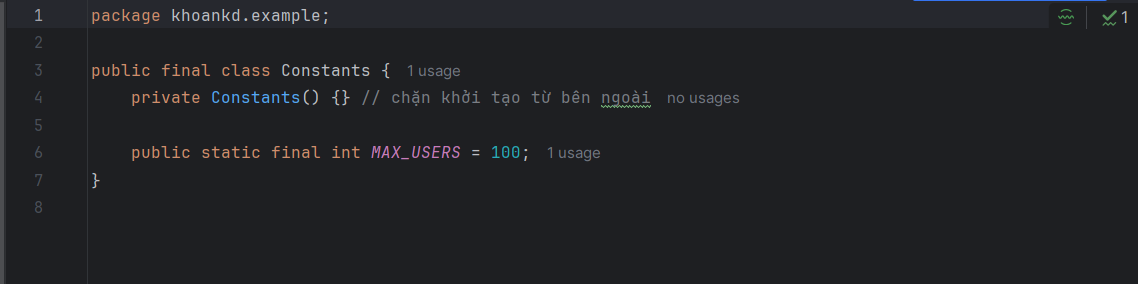
1. **InterfaceFieldModificationExample**

* Code ban đầu

****

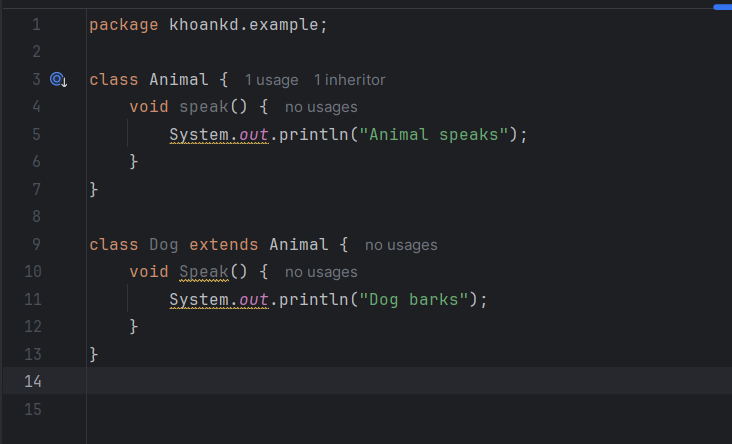
| Vị trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| Interface Constants | Interface chỉ chứa hằng số (MAX\_USERS) | Interface không nên chỉ dùng để chứa constant. Điều này không đúng với nguyên lý thiết kế module hóa và gây khó bảo trì về sau. | Chuyển sang final class Constants và dùng public static final đúng chuẩn |

* Code sạch và hết lỗi:



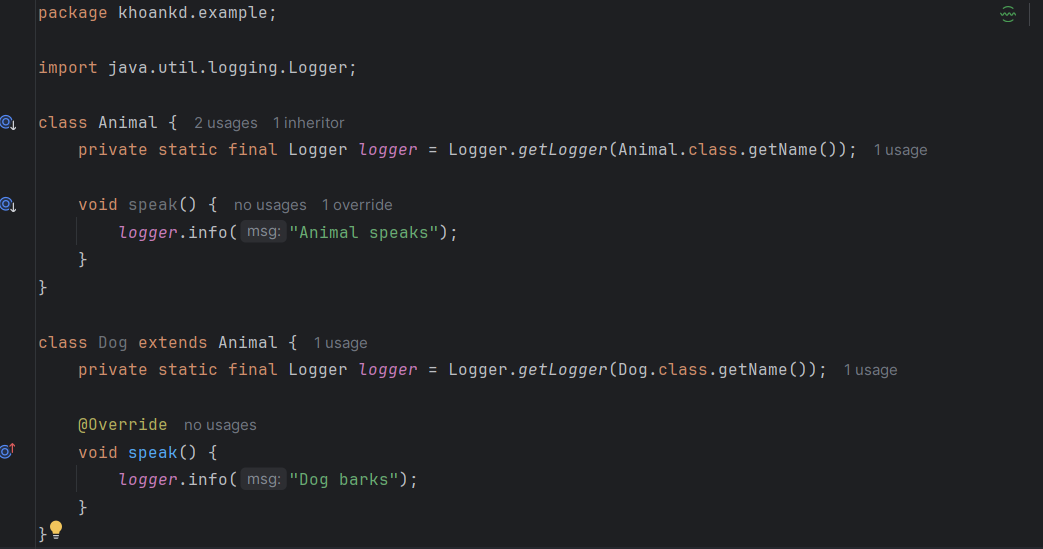
1. **MissingOverrideAnnotationExample**

* Code ban đầu:

****

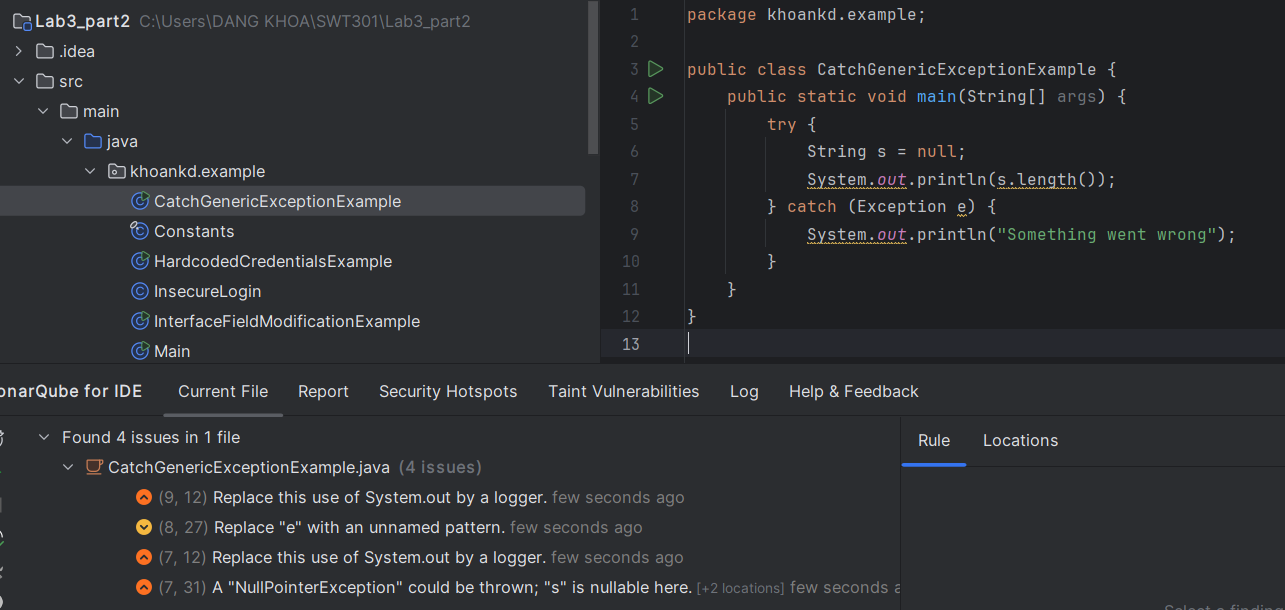
| * **Vị trí** | * **Lỗi** | * **Giải thích** | * **Cách sửa** |
| --- | --- | --- | --- |
| * Dòng 5 | * System.out.println trong Animal.speak() | * Không nên dùng System.out, gây khó theo dõi, không phân cấp độ log. | * Thay bằng Logger (logger.info(...)) |
| * Dòng 11 | * System.out.println trong Dog.Speak() | * Tương tự dòng 5 | * Dùng Logger |
| * Dòng 10 | * Tên phương thức Speak() không tuân theo chuẩn đặt tên Java | * Java khuyến nghị dùng camelCase (viết thường chữ đầu), tránh hiểu nhầm và không đồng nhất với override. | * Đổi tên Speak() thành speak() |
| * Dòng 10 | * Thiếu @Override khi override phương thức cha (speak()) | * Không có @Override sẽ khiến IDE và compiler không kiểm tra được override, dễ gây lỗi logic, khó bảo trì. | * Thêm @Override vào trước phương thức speak() |

* Code sạch và hết lỗi :



1. **CatchGenericExceptionExample**

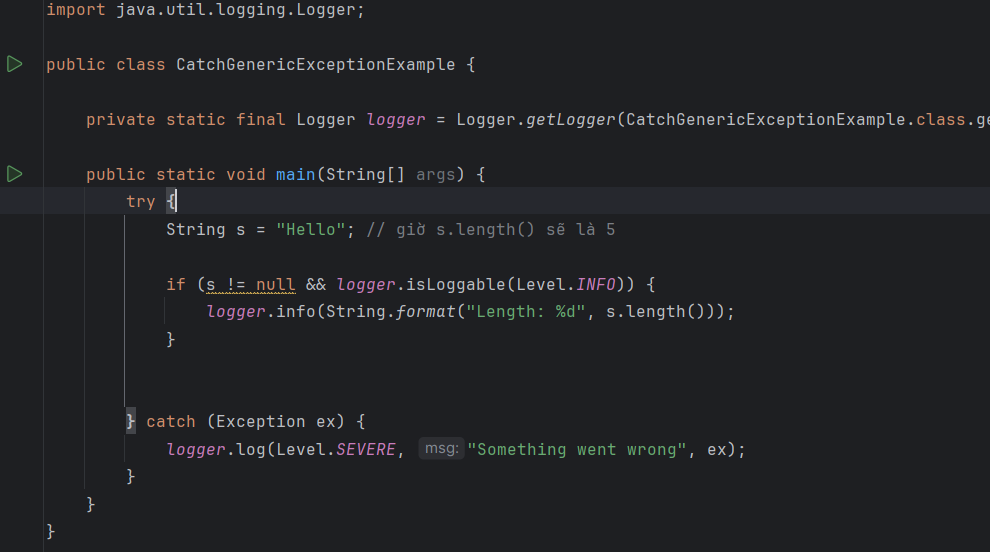
* Code ban đầu



-

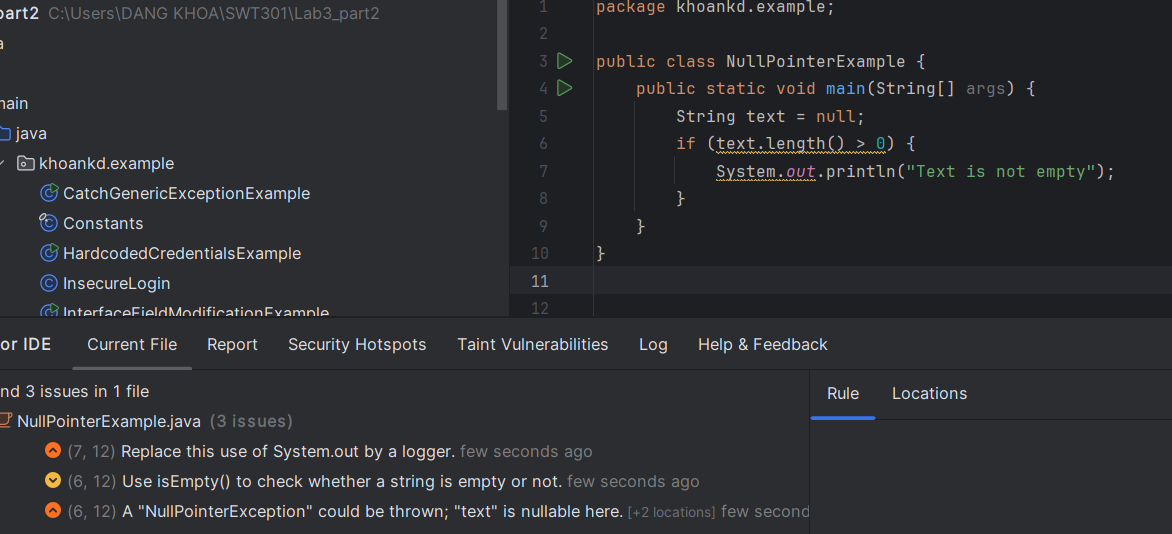
| **Vị trí** | **Lỗi** | **Giải thích** | **Cách sửa** |
| --- | --- | --- | --- |
| Dòng 7 | System.out.println(s.length()) | Không nên dùng System.out. Gây khó bảo trì, không có cấp độ log. | Thay bằng logger.info(...) |
| Dòng 9 | System.out.println("Something went wrong") | Tương tự lỗi trên | Dùng logger.severe(...) hoặc logger.warning(...) |
| Dòng 7 | "s" is nullable here | Biến s = null → gọi s.length() sẽ gây ra NullPointerException | Ghi chú rõ hoặc xử lý trước khi gọi |
| Dòng 8 | catch (Exception e) → dùng tên e mà không xử lý nó | Nếu không dùng e, có thể viết là catch (Exception ignored) hoặc \_ (Java 21+) để rõ ý định bỏ qua | Đổi tên biến hoặc ghi chú rõ ràng |

* Code sạch và hết lỗi



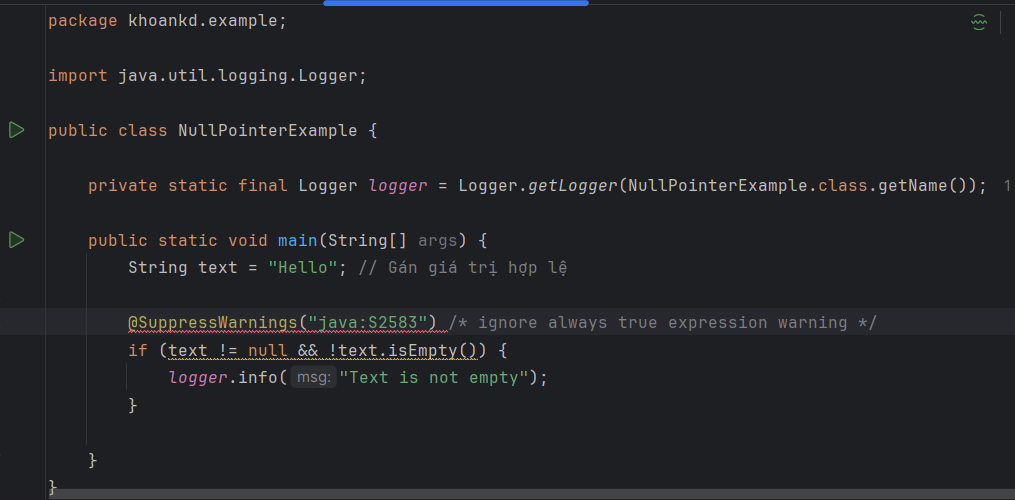
1. **NullPointerExample**

* Code ban đầu:

****

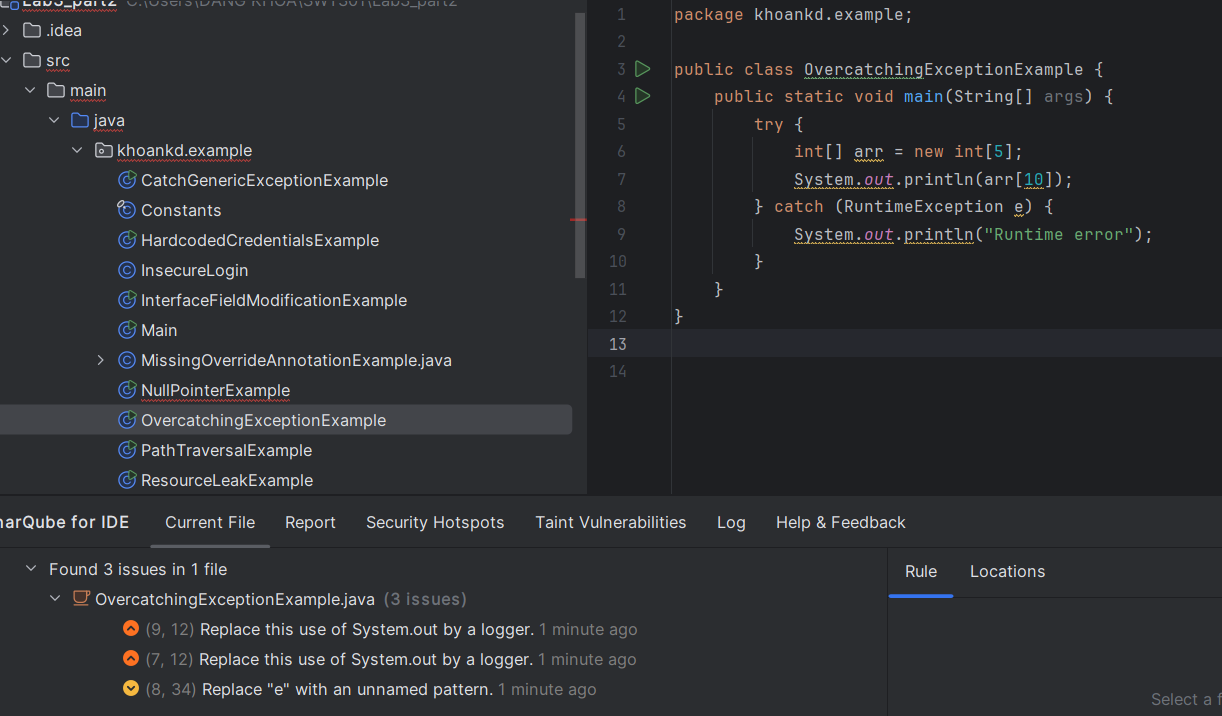
* Lỗi

| **Vị trí** | **Lỗi** | **Giải thích** | **Cách sửa** |
| --- | --- | --- | --- |
| Dòng 6 | text.length() → có thể gây NullPointerException | Vì text = null, nên gọi length() trực tiếp sẽ ném lỗi tại runtime | Kiểm tra text != null trước khi gọi .length() |
| Dòng 6 | Dùng isEmpty() thay vì text.length() > 0 | isEmpty() rõ ràng hơn, dễ đọc và ngắn gọn hơn | Dùng !text.isEmpty() thay vì text.length() > 0 |
| Dòng 7 | Không nên dùng System.out.println để log | Gây khó bảo trì, không phân cấp độ log (INFO, WARN, ERROR...) | Dùng Logger thay thế |

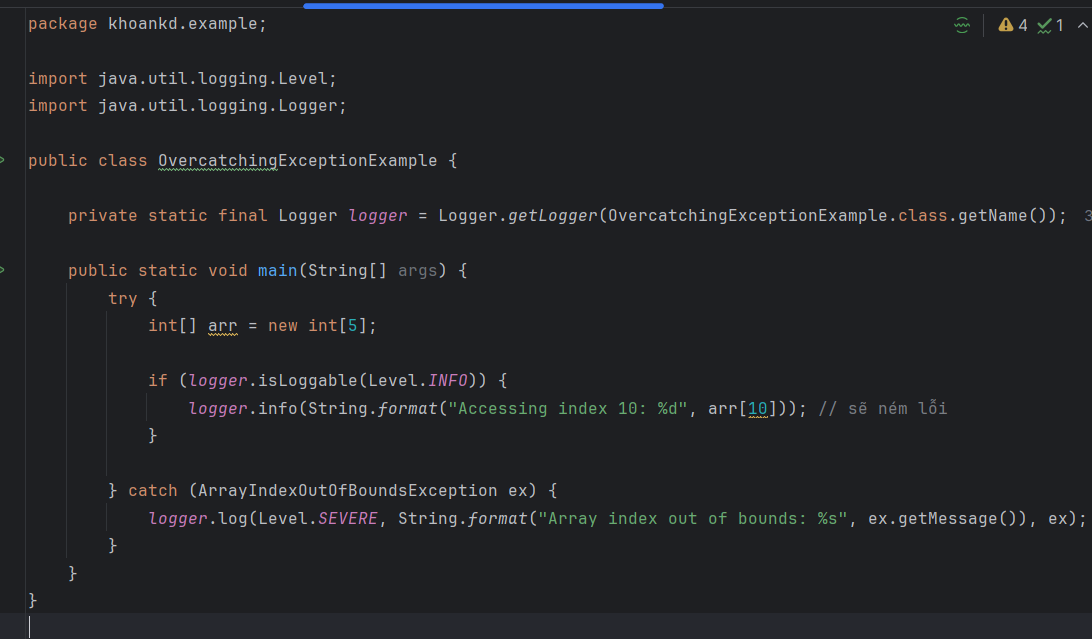


1. **OvercatchingExceptionExample**

* Code ban đầu



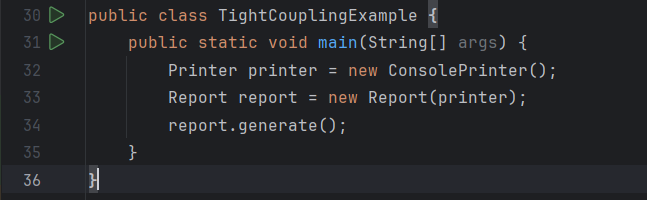
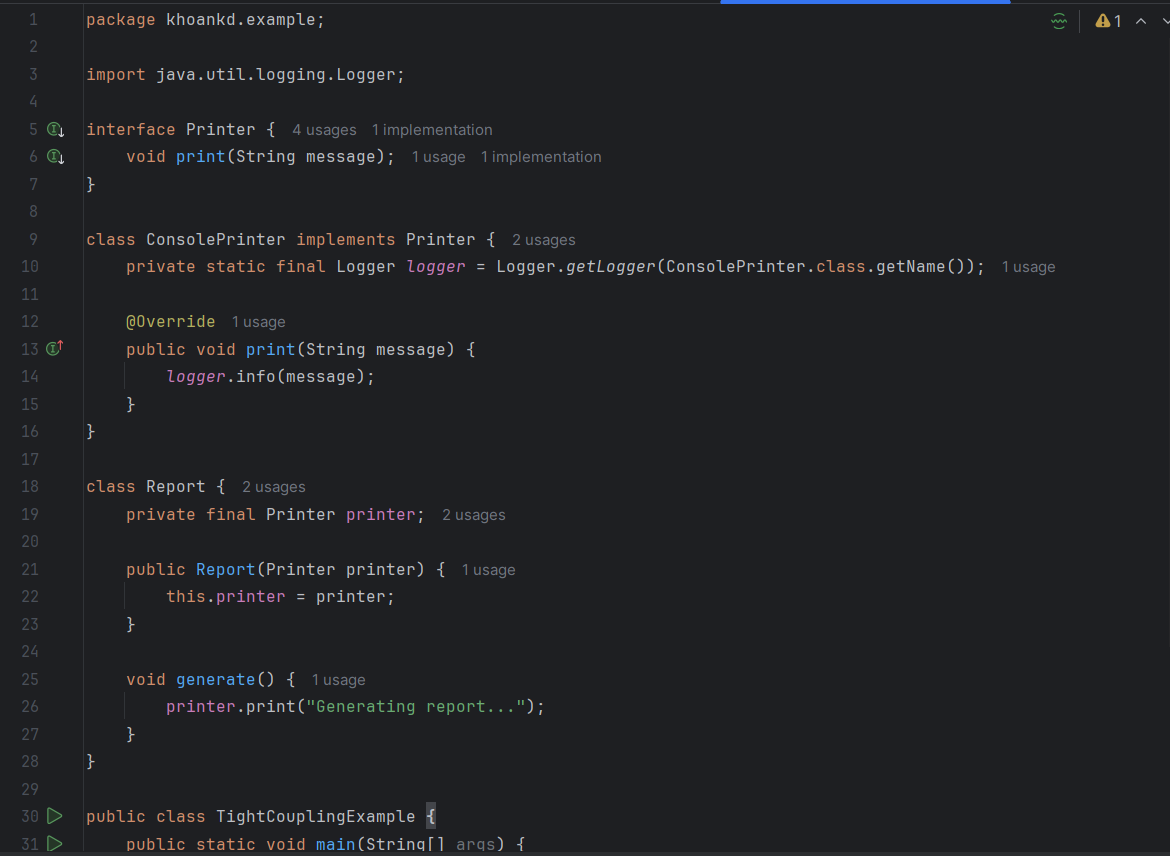




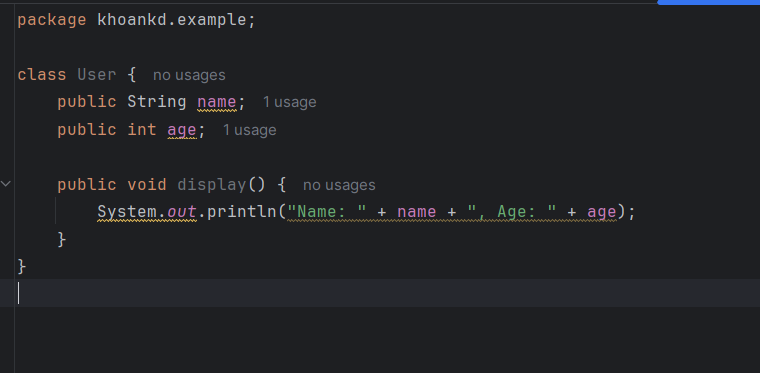
1. **TightCouplingExample**

****

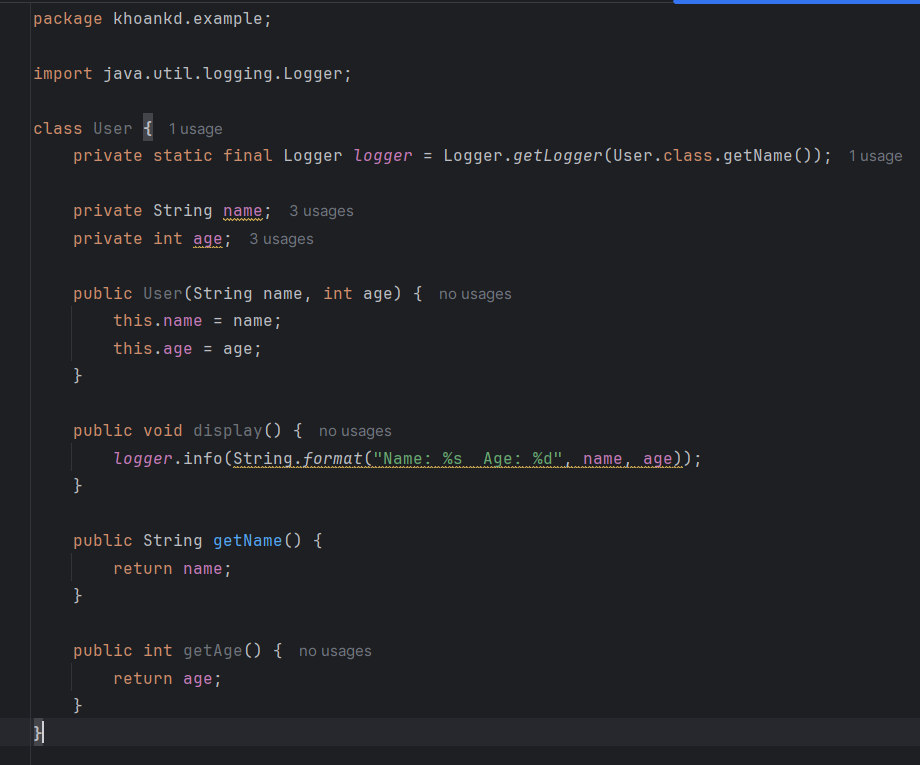
|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| Report khởi tạo Printer trực tiếp | Liên kết chặt (tight coupling) với lớp Printer | Report phụ thuộc vào cài đặt cụ thể thay vì giao diện, làm giảm tính mở rộng và test khó khăn | Tách giao diện Printer, tiêm phụ thông qua constructor (Dependency Injection) |



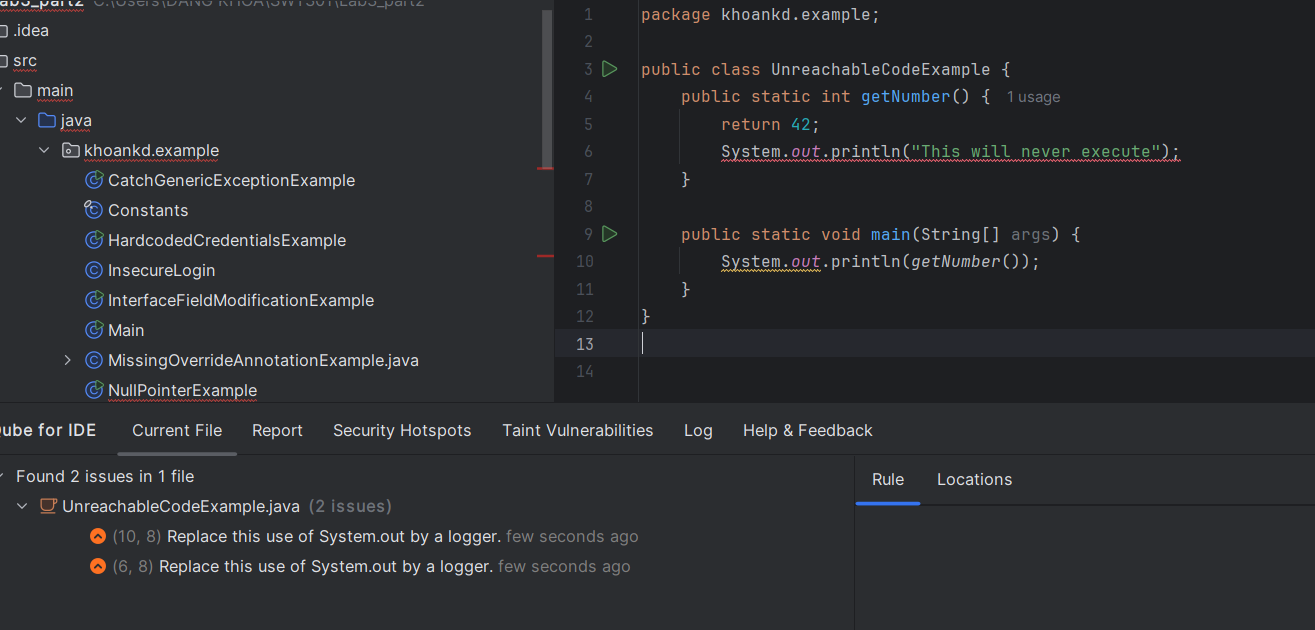
1. **ViolationOfEncapsulationExample**

****

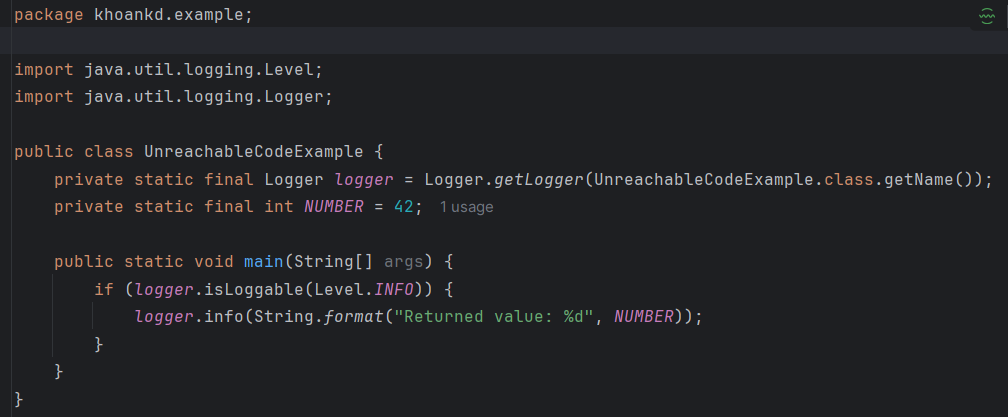
|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| System.out.println(...) trong display() | Không nên dùng để log thông tin | Không có phân loại cấp độ, khó kiểm soát log trong ứng dụng lớn | Dùng Logger thay vì System.out |

****

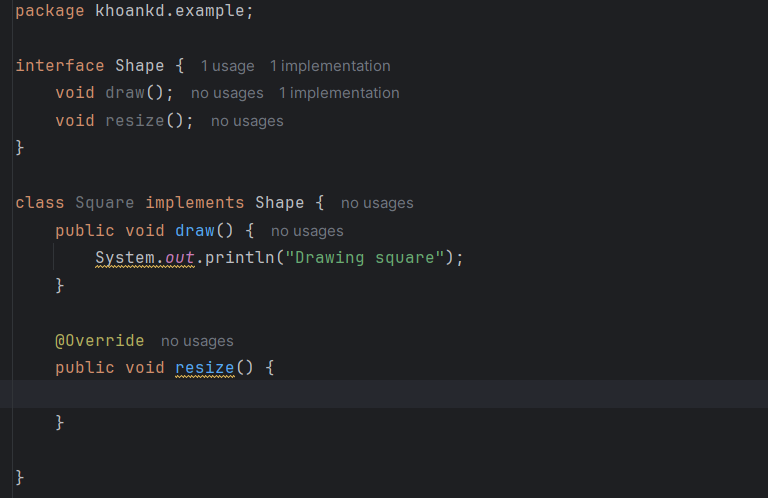
1. **UnreachableCodeExample**

****

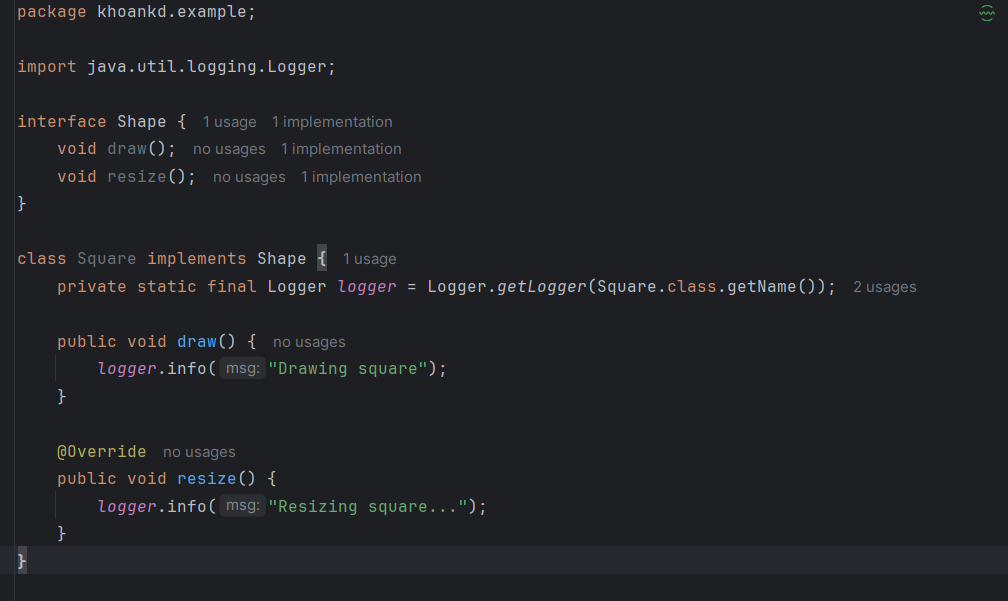
|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| Sau return | Dòng System.out.println(...) không bao giờ chạy | Gọi lệnh sau return là unreachable code → compiler warning hoặc lỗi logic | Xóa dòng không thể thực thi |
| System.out.println(...) | Không nên dùng để log (SonarLint) | Không phân cấp log, khó kiểm soát trong hệ thống lớn | Dùng Logger |

****

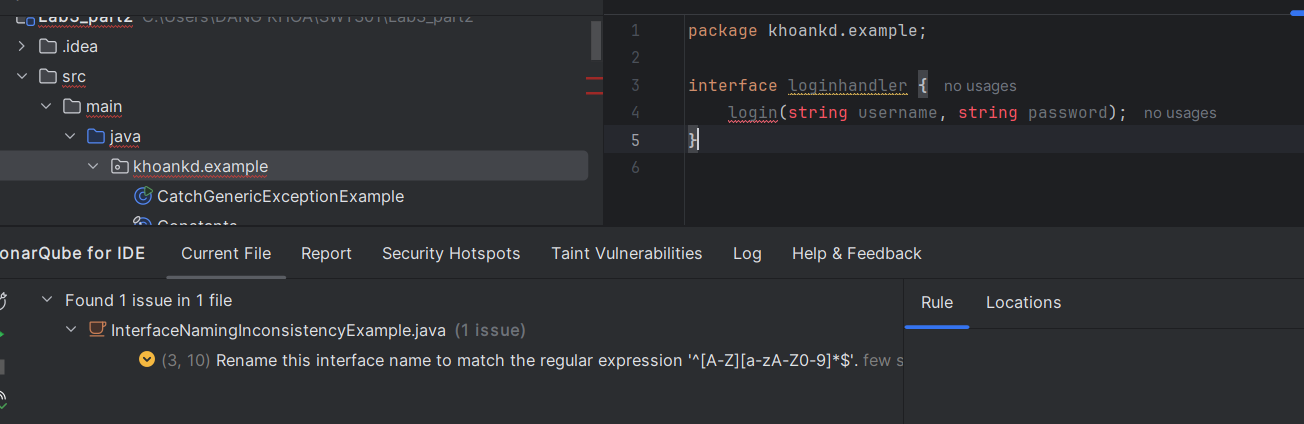
1. **IncompleteInterfaceImplementationExample**



|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| System.out.println(...) trong draw() | Không nên dùng log thô trực tiếp | Không có cấp độ log, không kiểm soát trong môi trường sản xuất | Dùng Logger thay vì System.out |
| resize() rỗng không có logic | Thiếu nội dung xử lý khi override | Gây hiểu nhầm về việc override nhưng không thực hiện hành động gì cụ thể | Viết thêm nội dung hợp lý hoặc log rằng chức năng đang được gọi |

****

1. **InterfaceNamingInconsistencyExample**



|  |  |  |  |
| --- | --- | --- | --- |
| Vị trí | Lỗi | Giải thích | Cách sửa |
| Tên interface loginhandler | Vi phạm quy tắc đặt tên interface (PascalCase) | Tên interface nên viết hoa chữ cái đầu và theo CamelCase để dễ đọc và nhất quán | Đổi tên thành LoginHandler, UserLoginHandler,... |

