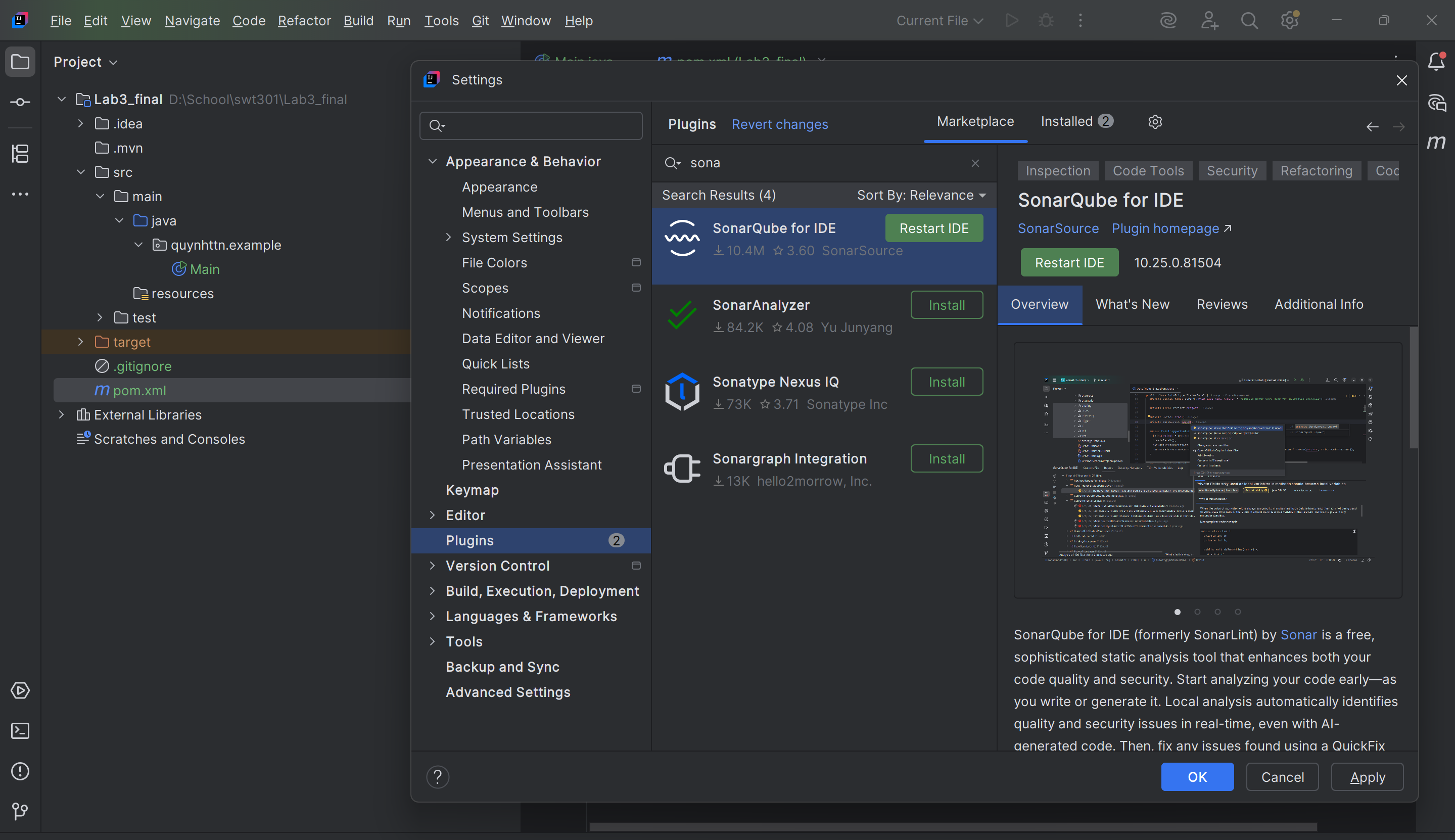
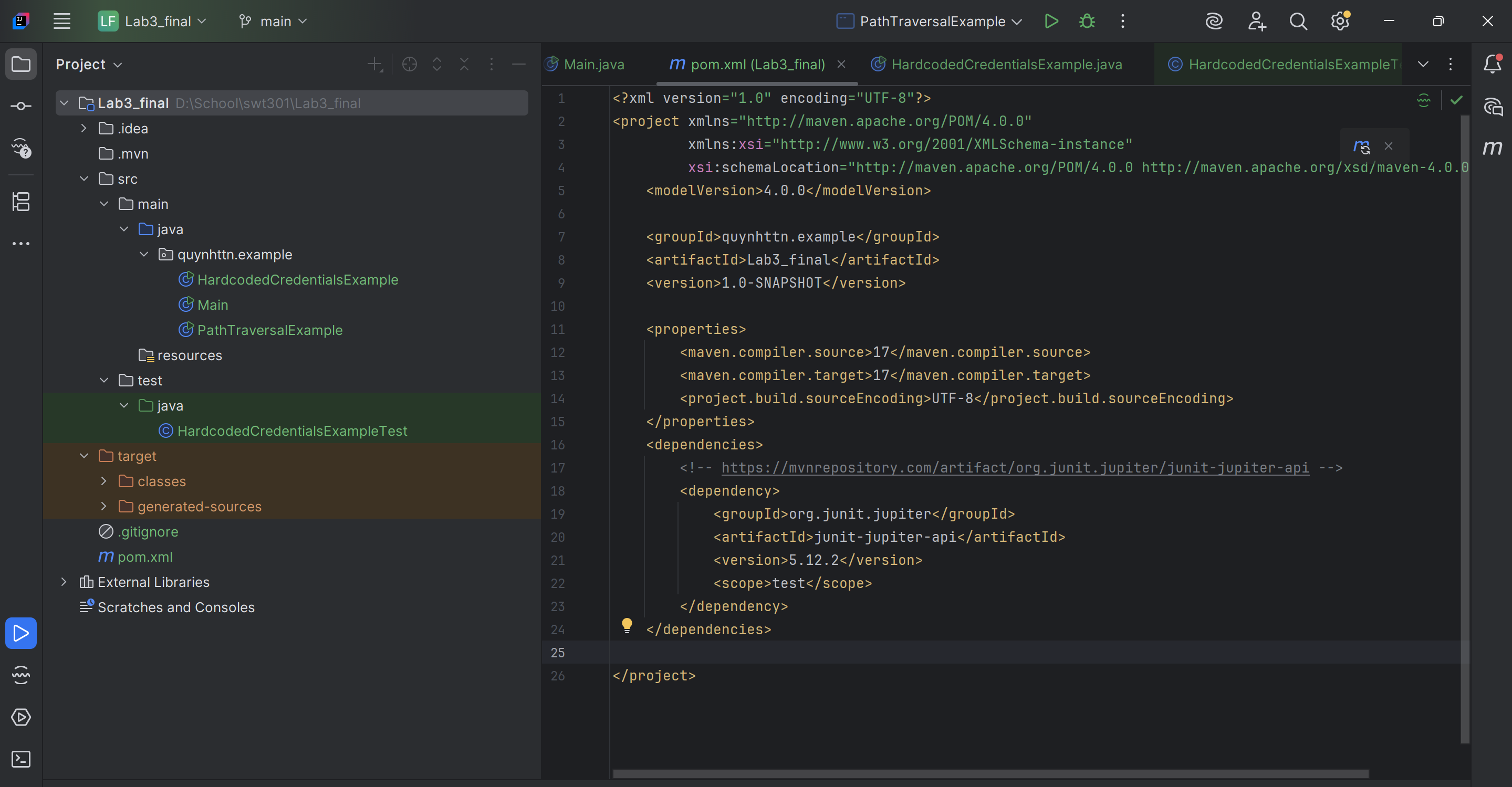
**Lab 3**

**Use AI for static code analysis**

# Tạo project mới và cài đặt plugin SonarQube:



# Copy các file java nguồn và tiến hành sử dụng công cụ SonarQube đã tích hợp vào Intellij đểchạy phân tích để phát hiện lỗi.

File pom.xml****

## HardcodedCredentialsExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.out.println("Access granted");(line 8) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| System.out.println("Access denied");(line 10) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |

### Viết lại code:

## PathTraversalExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.out.println("Access granted");(line 8) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| System.out.println("Access denied");(line 10) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |

### Viết lại code:

## PathTraversalExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println("Reading file: " + file.getPath());(line 11) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |

### Viết lại code:

## 3. ResourceLeakExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println(line);(line 10) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| new BufferedReader(new FileReader("data.txt"))(line 7) | Use try-with-resources or close this "BufferedReader" in a "finally" clause. | Không đóng tài nguyên, gây rò rỉ bộ nhớ và tài nguyên | Sử dụng try-with-resources để đảm bảo tài nguyên được đóng đúng cách |

### Viết lại code:

### SQLInjectionExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| **Vị trí** | **Lỗi** | **Giải thích** | **Cách sửa** |
| --- | --- | --- | --- |
| System.out.println("Executing query: " + query); (line 8) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| import java.sql.\*; (line 2) | Remove this unused import 'java.sql'. | Không cần thiết phải nhập 'java.sql' nếu không sử dụng | Dùng import chính xác thay vì nhập tất cả các lớp trong java.sql. |

### Viết lại code:

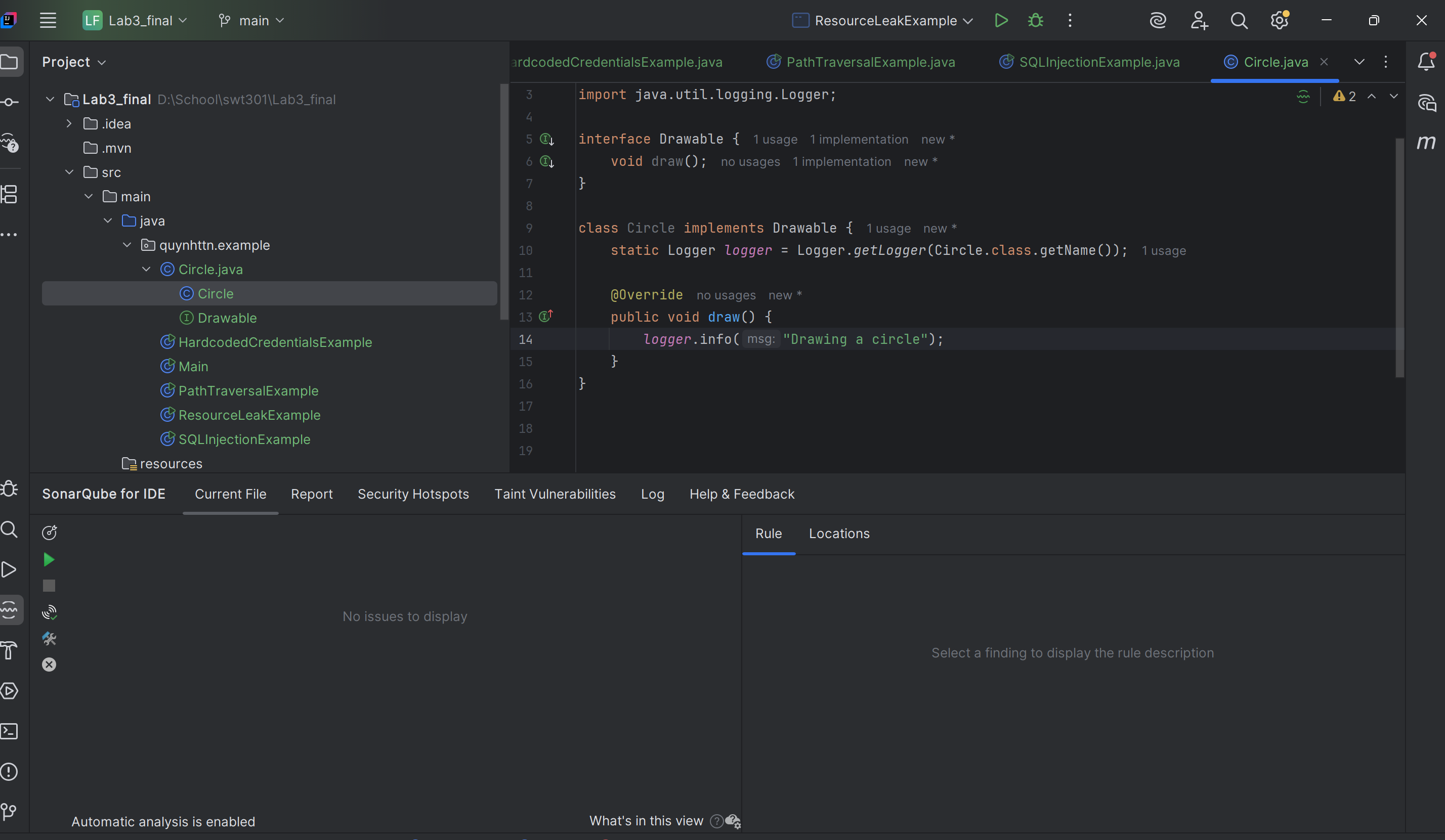
### 5.UnimplementedInterfaceExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| **Vị trí** | **Lỗi** | **Giải thích** | **Cách sửa** |
| --- | --- | --- | --- |
| class Circle implements Drawable { // Missing draw() implementation(line 7) | Unimplemented interface method draw() | Lỗi xảy ra do lớp Circle không triển khai phương thức draw() được khai báo trong interface Drawable. | Triển khai phương thức draw() trong lớp Circle. |

### Viết lại code:



### 6.InterfaceFieldModificationExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| interface Constants {  int *MAX\_USERS* = 100; };(line 3) | Move constants defined in this interfaces to another class or enum | Các trường hợp trong interface không thể thay đổi giá trị của biến vì các trường hợp này được khai báo như là final mặc định. | Không sửa giá trị của trường trong interface. |

### Viết lại code:

### 7.MissingOverrideAnnotationExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| void Speak() {;(line 10) | Rename method "Speak" to prevent any misunderstanding/clash with method "speak" defined in superclass | Phương thức Speak() trong lớp Dog không ghi đè chính xác phương thức speak() của lớp cha Animal. Việc này có thể gây nhầm lẫn vì nó không phải là ghi đè đúng. | Đổi tên thành speak() |
| void Speak() {;(line 10) | Add the "@Override" annotation above this method signature | Ghi đè phải có override | Thêm override |
| System.*out*.println("Animal speaks");line 5System.out.println("Dog barks");(line11) | Replace this use of System.out by a logger. | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |

### Viết lại code:

### 8.CatchGenericExceptionExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println(s.length());(line 7) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| System.*out*.println("Something went wrong");(line 9) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| s.length() line 7; | A "NullPointerException" could be thrown; "s" is nullable here. | Null không tồn tại phương thức để gọi length | Thêm if s!= null |

### Viết lại code:

### 9.NullPointerExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println("Text is not empty");line 7) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| if (text.length() > 0) {(line 6) | Use isEmpty() to check whether a string is empty or not. | Khi biến text là null, gọi text.length() sẽ gây ra ngoại lệ NullPointerException. | Kiểm tra text không phải là null trước khi gọi phương thức length(). |
| if (text.length() > 0) {(line 6) | A "NullPointerException" could be thrown; "text" is nullable here. | Khi biến text là null, gọi text.length() sẽ gây ra ngoại lệ NullPointerException. | Kiểm tra text không phải là null trước khi gọi phương thức length(). |

### Viết lại code:

### 10.OvercatchingExceptionExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println(arr[10]);(line 7)System.*out*.println("Runtime error");(line 9) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |

### Viết lại code:

### 11.TightCouplingExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println(message);(line 5) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |

### Viết lại code:

### 12.ViolationOfEncapsulationExample

### Code nguồn ban đầu:

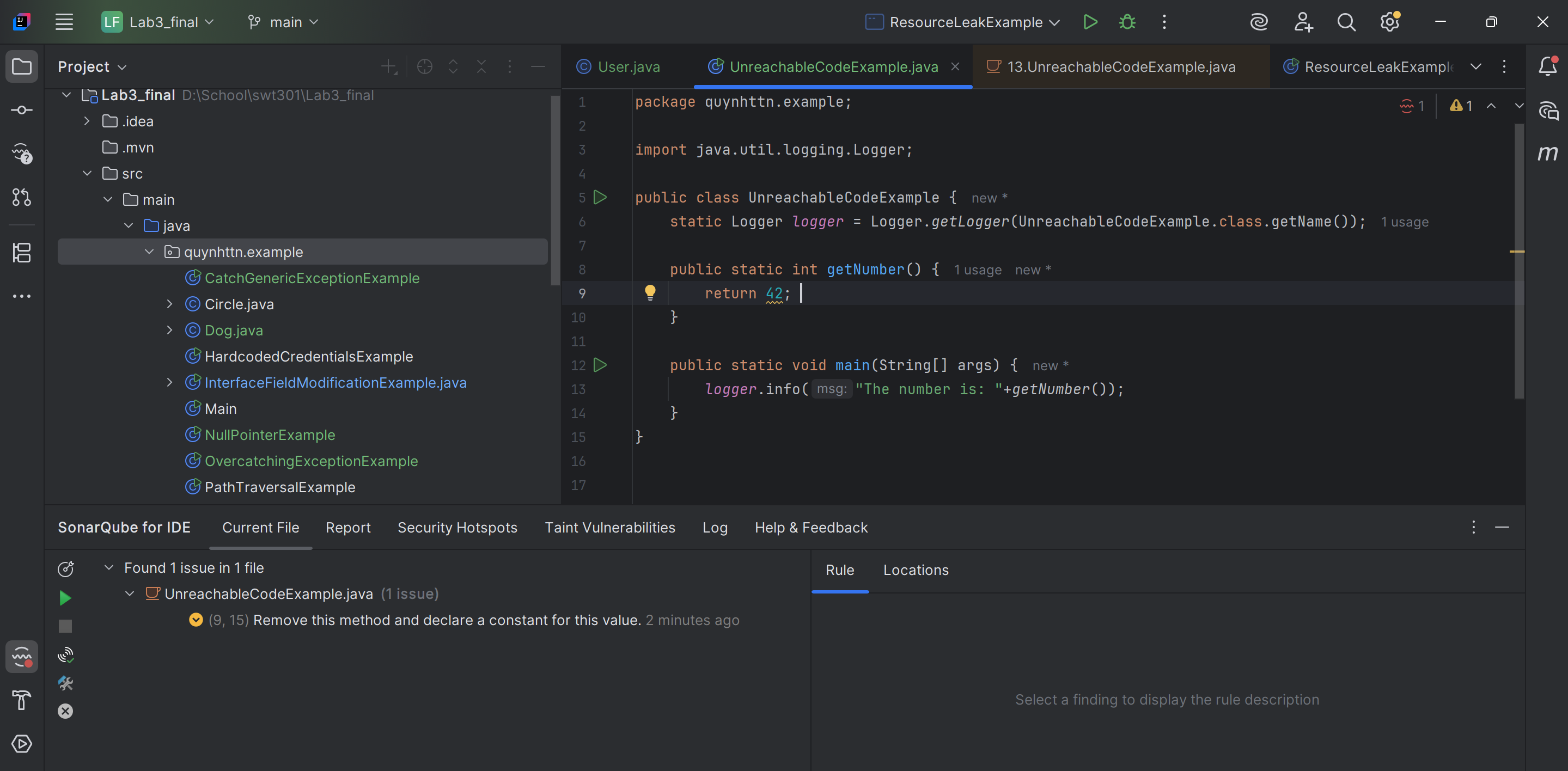
### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println("Name: " + name + ", Age: " + age);(line 8) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| public String name; | Make name a static final constant or non-public and provide accessors if needed. | Vi phạm nguyên tắc đóng gói trong lập trình hướng đối tượng. | Đổi thành private |
| public int age; | Make age a static final constant or non-public and provide accessors if needed. | Vi phạm nguyên tắc đóng gói trong lập trình hướng đối tượng. | Đổi thành private |

### Viết lại code:

### 13.UnreachableCodeExample

### Code nguồn ban đầu:



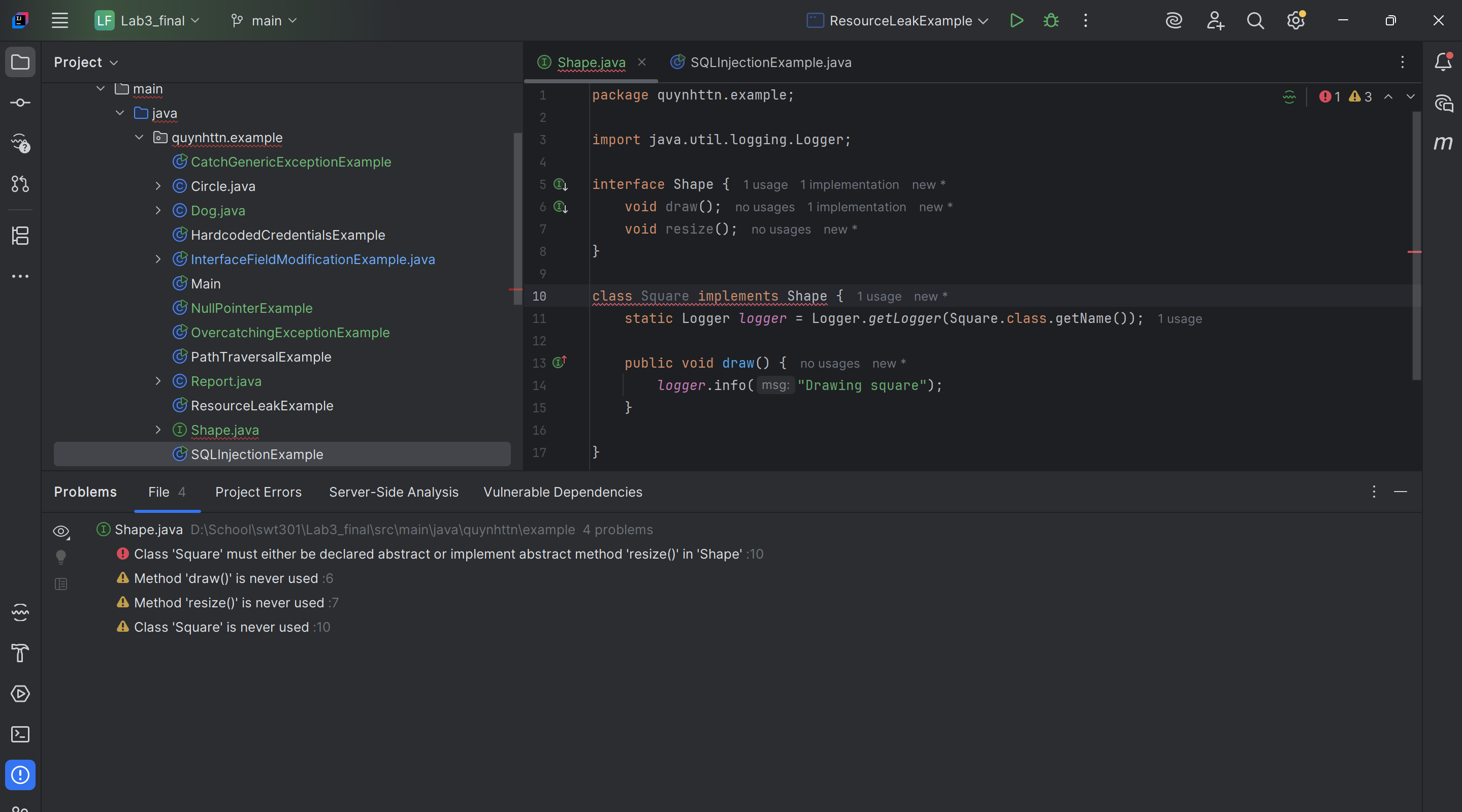
### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println("This will never execute");(line 6) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| System.*out*.println(*getNumber*());(line 8) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| return 42; | Remove this method and declare a constant for this value. |  |  |

### Viết lại code:

### 14.IncompleteInterfaceImplementationExample

### Code nguồn ban đầu:



### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| System.*out*.println("Drawing square");(line 10) | Replace this use of System.out by a logger | Không nên dùng System.out trong ứng dụng thực, thay bằng logging framework như java.util.logging, Log4j, hoặc SLF4J | Dùng Logger thay thế. |
| class Square implements Shape (line 8) | Class 'Square' must either be declared abstract or implement abstract method 'resize()' in 'Shape | Không override method resize() của interface Shape | Thêm override method resize() |

### Viết lại code:

## 15.InterfaceNamingInconsistencyExample

### Code nguồn ban đầu:

### Phân tích lỗi và giải thích:

| Vi trí | Lỗi | Giải thích | Cách sửa |
| --- | --- | --- | --- |
| interface loginhandler(line 3) | Rename this interface name to match the regular expression '^[A-Z][a-zA-ZO-9]\*$'. | Sai quy tắc đặt tên | Dùng Logger thay thế. |
| login(string username, string password);(line 4) | Invalid method declaration; return type required :4 | Thiếu kiểu dữ liệu trả về | Thêm kiểu dữ liệu trả về |
| login(string username, string password);(line 4) | Cannot resolve symbol 'string' : | Sai kiểu dữ liệu | Đổi thành String |

### Viết lại code: