

Java 8 - Functional Interface

Functional Interface là một interface có một phương thức abstract, nó cũng có thể được gọi là Single Abstract Interface (SAM) một cụm từ đôi khi chúng ta bắt gặp.

`@FunctionalInterface` annotation được thêm vào để chúng ta đánh dấu interface đó là functional interface, điều này không bắt buộc nhưng có thể là cách tốt nhất trong việc sử dụng functional interface để tránh vô tình thêm các phương thức khác. Nếu một interface đã có annotation `@FunctionalInterface` và chúng ta cố gắng thêm vào các phương thức khác thì trình biên dịch sẽ ném ra lỗi.

Lợi ích chính của functional interface là chúng ta có thể sử dụng Lambda Expression để tạo ra thể hiện (instance) cho interface đó.

Ví dụ:

```
@FunctionalInterface
public interface Functional {
    void method();
}
```

Một điều cần lưu ý: phương thức default and static không phá vỡ quy tắc của functional interface.

Ví dụ:

```
@FunctionalInterface
public interface FunctionalDefaultMethods {
    void method();

    default void defaultMethod() {
    }
}
```

Ví dụ: Trong ví dụ sau, chúng ta sẽ gán biểu thức lambda cho functional interface. Sau đó chúng ta có thể chạy biểu thức lambda bằng cách gọi phương thức được định nghĩa bên trong functional interface và truyền tham số vào.

```
public class Main {
    public static void main(String[] argv) {
        Processor stringProcessor = (String str) -> str.length();
    }
}
```

```
String name = "Java Lambda";
int length = stringProcessor.getStringLength(name);
System.out.println( length);
}

@FunctionalInterface
interface Processor {
    int getStringLength(String str);
}
```

Đoạn code trên trả về kết quả như sau: 11

Revision #1

Created 29 September 2019 16:23:52 by Laptrinh.vn

Updated 19 October 2019 16:55:59 by Laptrinh.vn