Question: Action Event Handler

Problem Statement:

You are tasked with creating a simple event handling system for a user interface. You need to define an enumeration for different types of actions that can be performed in the application. Additionally, you will create an interface that defines a method for handling these actions.

1. Create an enumeration called **ActionEnum** that includes the following actions:
    - **CREATE**
    - **READ**
    - **UPDATE**
    - **DELETE**
2. Create an interface called **IActionEvent** that declares a method:
    - **void handleAction(ActionEnum action);**
3. Implement the **IActionEvent** interface in a class called **ActionHandler**.
   The **handleAction** method should print a message indicating which action is being handled.
4. In the main method, create a list of **ActionEnum** values and iterate through the list, passing each action to an instance of **ActionHandler**.

1Handling action: CREATE

2Handling action: READ

3Handling action: UPDATE

4Handling action: DELETE

Solution:

Here is a possible implementation in Java:

java

VerifyOpen In EditorRunCopy code

```
1import java.util.ArrayList;
2import java.util.List;
3
4// Step 1: Define the ActionEnum
5enum ActionEnum {
6    CREATE,
7    READ,
```

```
8    UPDATE,
9    DELETE
10}
11
12// Step 2: Define the IActionEvent interface
13interface IActionEvent {
14    void handleAction(ActionEnum action);
15}
16
17// Step 3: Implement the IActionEvent interface in ActionHandler
18class ActionHandler implements IActionEvent {
19    @Override
20    public void handleAction(ActionEnum action) {
21        System.out.println("Handling action: " + action);
22    }
23}
24
25// Main class to demonstrate the functionality
26public class Main {
27    public static void main(String[] args) {
28        // Step 4: Create a list of ActionEnum values
29        List<ActionEnum> actions = new ArrayList<>();
30        actions.add(ActionEnum.CREATE);
31        actions.add(ActionEnum.READ);
32        actions.add(ActionEnum.UPDATE);
33        actions.add(ActionEnum.DELETE);
34
35        // Create an instance of ActionHandler
36        IActionEvent actionHandler = new ActionHandler();
37
38        // Iterate through the list and handle each action
```

```
39      for (ActionEnum action : actions) {
40          actionHandler.handleAction(action);
41      }
42  }
43}
```

Explanation:

1. **Enumeration**: The **ActionEnum** enum defines four actions: **CREATE**, **READ**, **UPDATE**, and **DELETE**.
2. **Interface**: The **IActionEvent** interface declares a method **handleAction(ActionEnum action)** that takes an **ActionEnum** as a parameter.
3. **Implementation**: The **ActionHandler** class implements the **IActionEvent** interface and provides the logic for handling actions by printing a message.
4. **Main Method**: In the **main** method, a list of **ActionEnum** values is created, and each action is passed to the **ActionHandler** instance for processing.