**Exercise 9: Implementing the Command Pattern**

**Scenario:** You are developing a home automation system where commands can be issued to turn devices on or off. Use the Command Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **CommandPatternExample**.
2. **Define Command Interface:**
   * Create an interface Command with a method **execute()**.
3. **Implement Concrete Commands:**
   * Create classes **LightOnCommand**, **LightOffCommand** that implement Command.
4. **Implement Invoker Class:**
   * Create a class **RemoteControl** that holds a reference to a Command and a method to execute the command.
5. **Implement Receiver Class:**
   * Create a class **Light** with methods to turn on and off.
6. **Test the Command Implementation:**
   * Create a test class to demonstrate issuing commands using the **RemoteControl**.

**Command Interface**: The Command interface defines the execute method that all command classes must implement.

**LightOnCommand Class**: This class implements the Command interface and provides the implementation for turning on the light.

**LightOffCommand Class**: This class implements the Command interface and provides the implementation for turning off the light.

**RemoteControl Class**: This class holds a reference to a Command and has a setCommand method to set the current command and a pressButton method to execute the current command.

**Light Class**: This class has methods to turn on and off the light. It acts as the receiver in the Command pattern.

**Testing the Command Pattern**: In the CommandPatternTest class, we create instances of Light, LightOnCommand, LightOffCommand, and RemoteControl. We set different commands at runtime and execute them using the remote control.