**Exercise 6: Implementing the Proxy Pattern**

**Scenario:**

You are developing an image viewer application that loads images from a remote server. Use the Proxy Pattern to add lazy initialization and caching.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **ProxyPatternExample**.
2. **Define Subject Interface:**
   * Create an interface Image with a method **display()**.
3. **Implement Real Subject Class:**
   * Create a class **RealImage** that implements Image and loads an image from a remote server.
4. **Implement Proxy Class:**
   * Create a class **ProxyImage** that implements Image and holds a reference to RealImage.
   * Implement lazy initialization and caching in **ProxyImage**.
5. **Test the Proxy Implementation:**
   * Create a test class to demonstrate the use of **ProxyImage** to load and display images.

**Image Interface**: The Image interface defines the display method that all image classes must implement.

**RealImage Class**: The RealImage class implements the Image interface and simulates loading an image from a remote server. It prints messages to simulate loading and display operations.

**ProxyImage Class**: The ProxyImage class implements the Image interface and holds a reference to RealImage. It uses lazy initialization to load the image from the remote server only when it is first requested. Subsequent requests to display the image use the cached RealImage instance.

**Testing the Proxy Pattern**: In the ProxyPatternTest class, we create instances of ProxyImage for different images. The first time an image is displayed, it is loaded from the remote server. Subsequent display requests for the same image use the cached instance, demonstrating lazy initialization and caching.