**Exercise 6: Implementing the Proxy Pattern**

**Image.java**

public interface Image {

void display();

}

**ReaLImage.java**

public class ReaLImage implements Image {

private String fileName;

public ReaLImage(String fileName) {

this.fileName = fileName;

loadFromRemoteServer();

}

private void loadFromRemoteServer() {

System.***out***.println("Loading image from remote server: " + fileName);

}

*@Override*

public void display() {

System.***out***.println("Displaying image: " + fileName);

}

}

**ProxyImage.java**

public class ProxyImage implements Image {

private String fileName;

private ReaLImage realImage;

public ProxyImage(String fileName) {

this.fileName = fileName;

}

*@Override*

public void display() {

if (realImage == null) {

realImage = new ReaLImage(fileName); // Lazy loading

}

realImage.display(); // Cached display

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

Image image1 = new ProxyImage("nature\_photo.jpg");

Image image2 = new ProxyImage("profile\_pic.png");

// Image will be loaded from server only when display is called

System.*out*.println("\nFirst time viewing nature\_photo.jpg:");

image1.display(); // Loads and displays

System.*out*.println("\nSecond time viewing nature\_photo.jpg:");

image1.display(); // Only displays (cached)

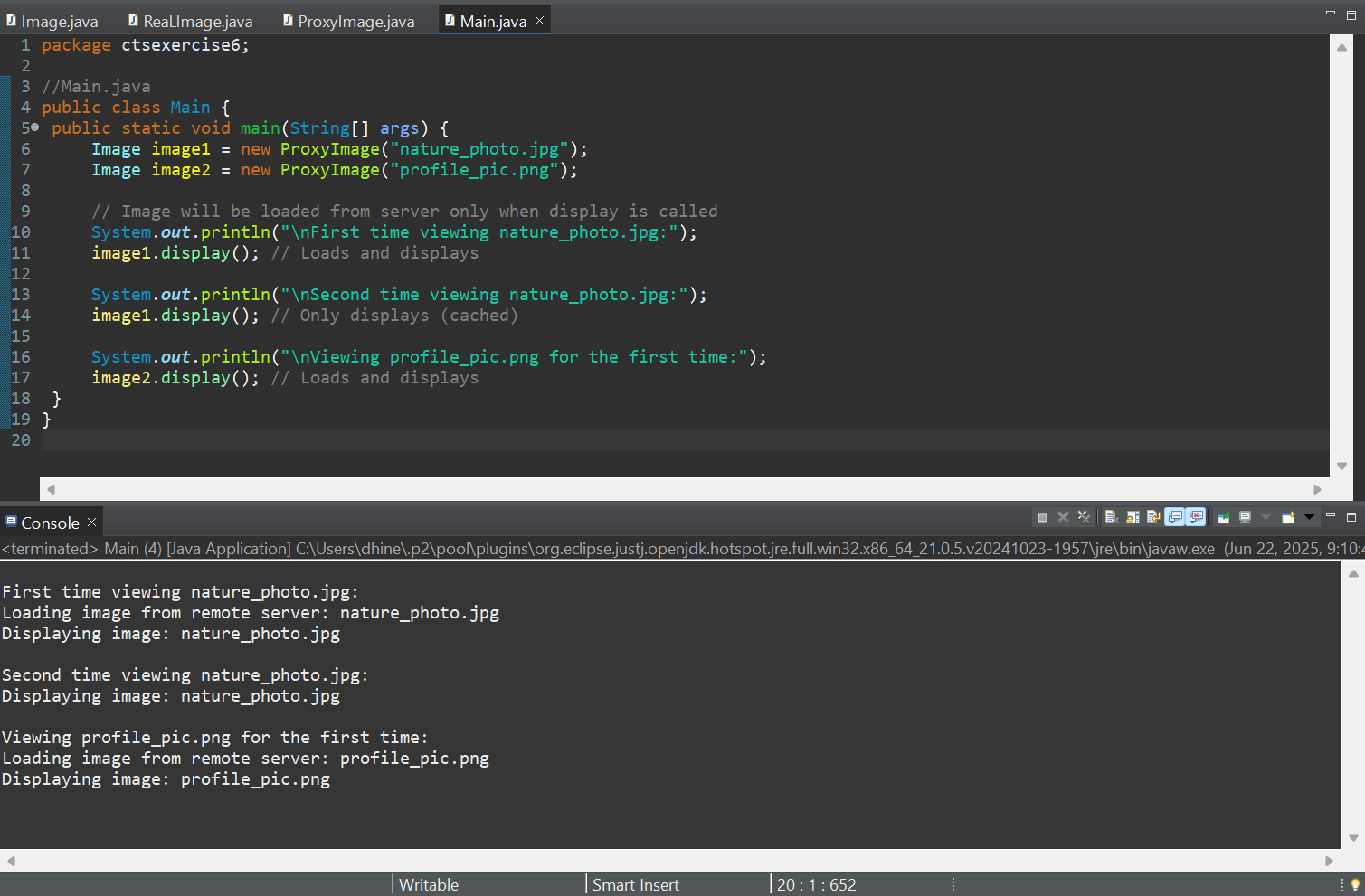
System.*out*.println("\nViewing profile\_pic.png for the first time:");

image2.display(); // Loads and displays

}

}

**Output:**

****