Strategy Pattern

In this lab, you will invoke two encryption strategies.

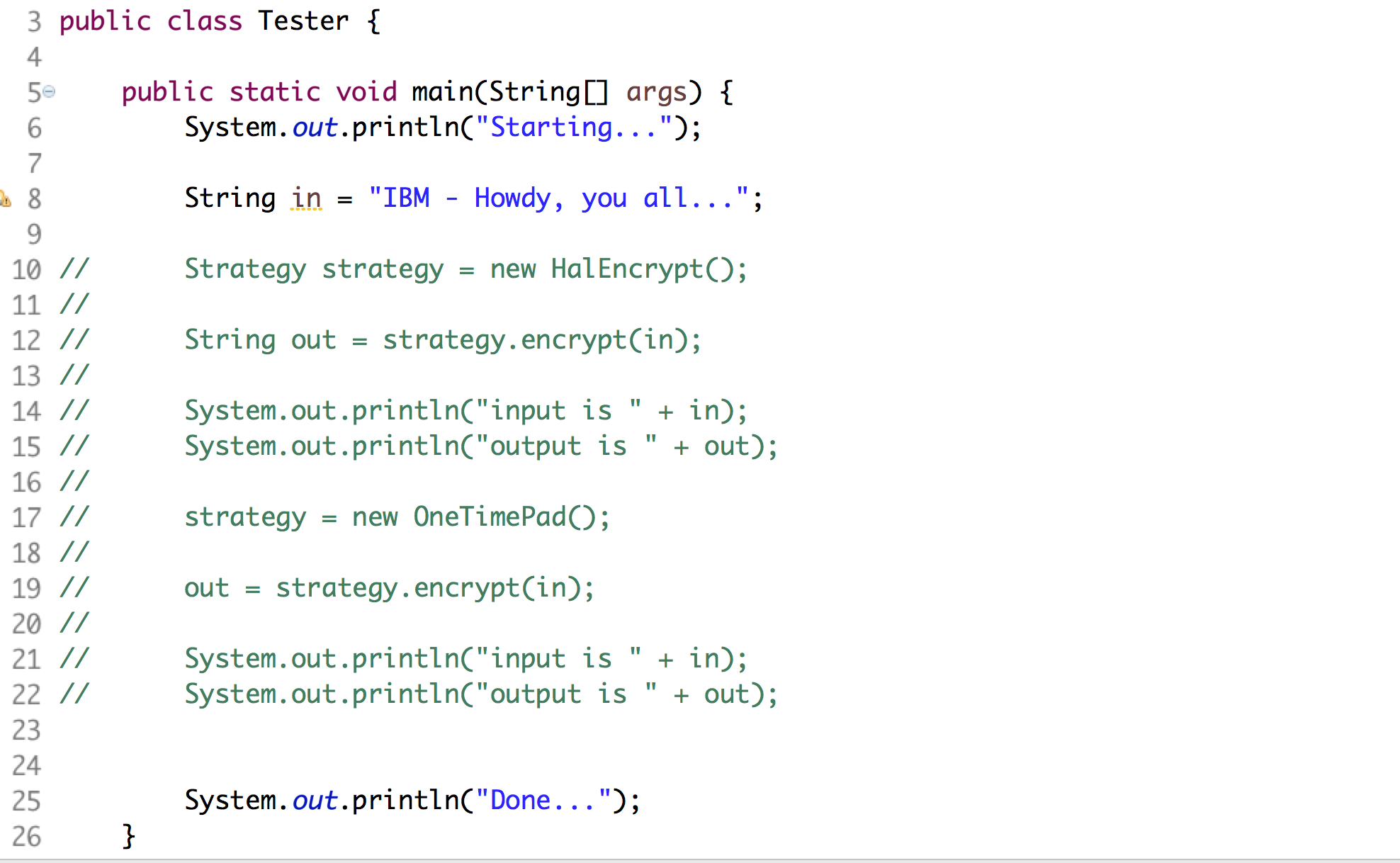
# Objectives

In this lab, you will

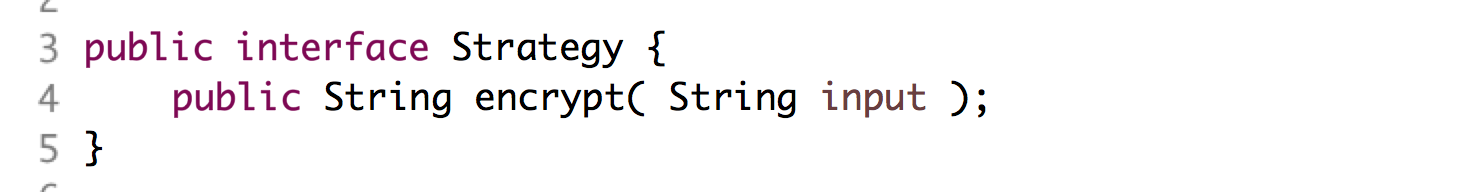
* Examine the Strategy interface
* Examine two encryption Strategy implementations
* Invoke the two encryption stategies

# Exercise

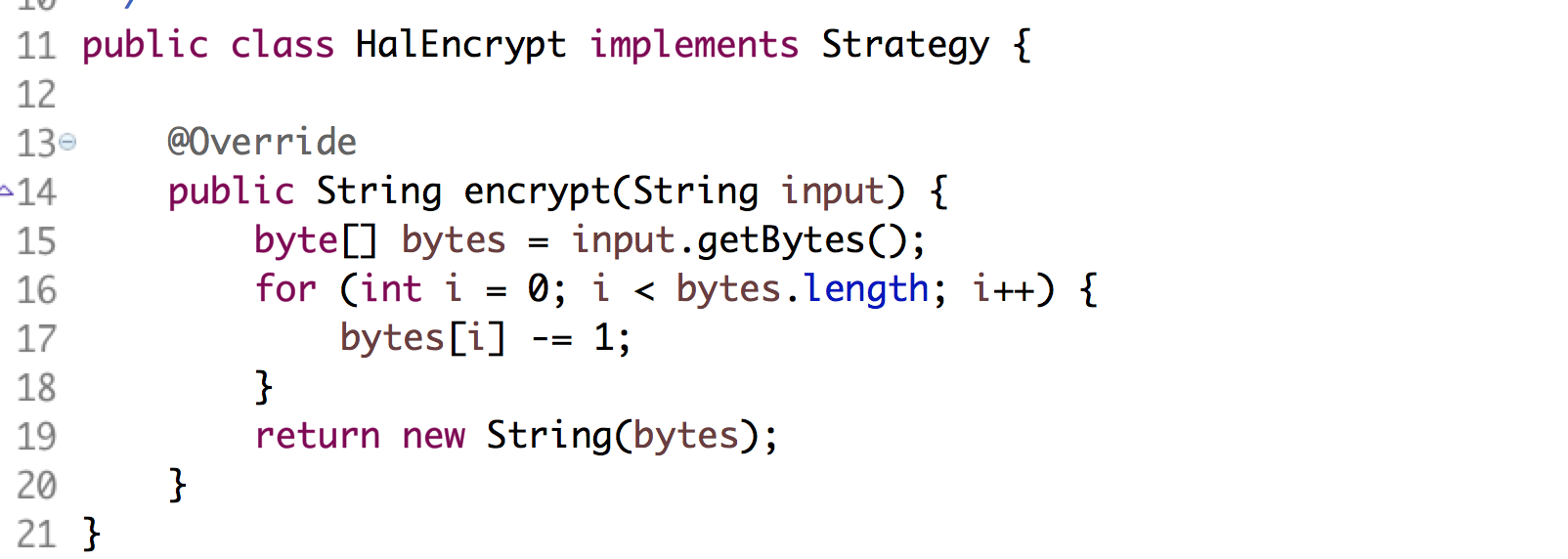
1. The Strategy Pattern represents multiple ways (or strategies) for accomplishing a task. The advantage is that it decouples the methods from the client.
2. In this example, we have two different ways to encrypt text. We call each one and display the result. NOTE: in the real world the client would call a Factory to get the strategy it wanted.
3. In Eclipse, in the exercises project, open the package com.paypal.patterns.Strategy to view the project files.
4. Examine the client, Tester.java, shown below:



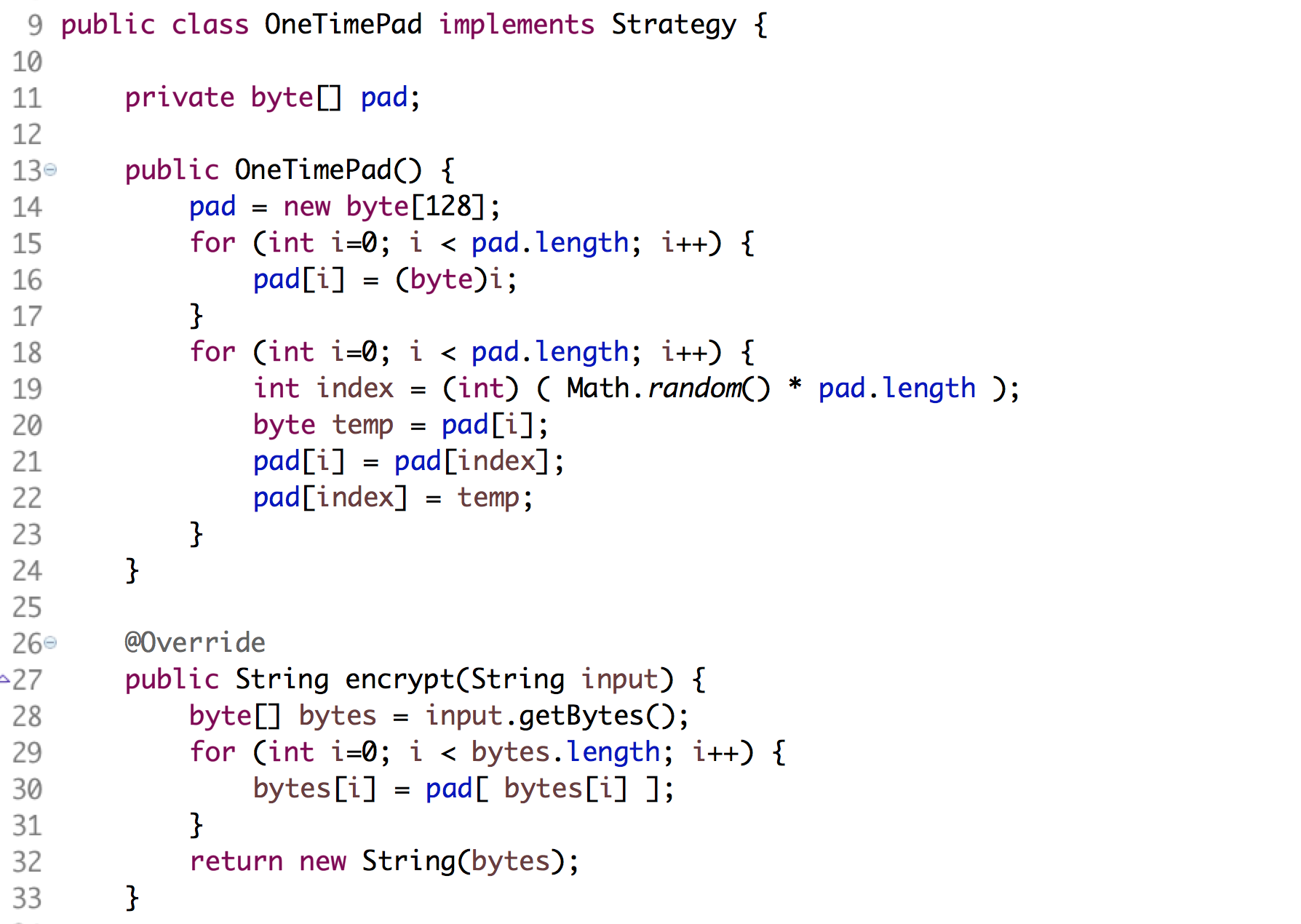
1. From lines 10-15, it uses the HalEncrypt strategy to encrypt the text. In the movie, “2001, A Space Odyssey”, the computer was named the HAL 9000. HAL is IBM with each letter smaller than the original. The HalEncrypt does the same thing. This is me being clever!!!
2. Examine the Strategy.java shown below:



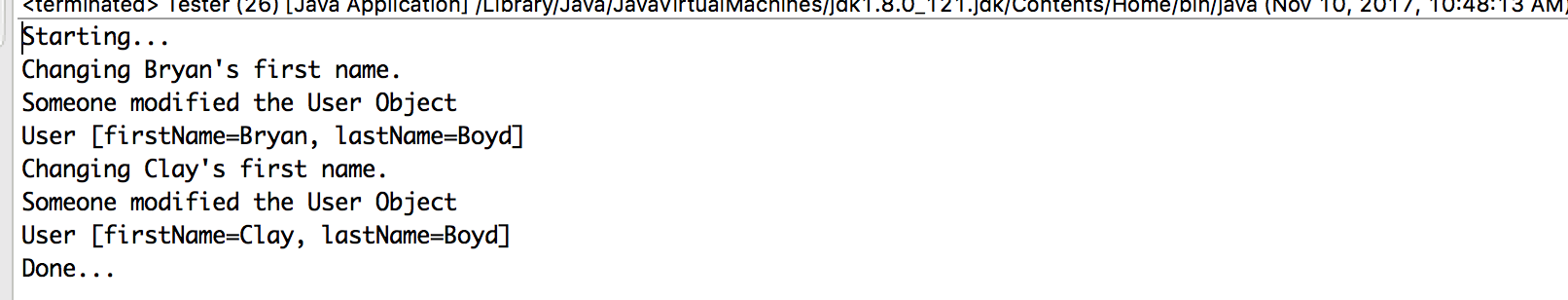
1. The above defines the method, encrypt(). The realizations of Strategy define different ways to do the same thing, encrypt a string.
2. Examine the HalEncrypt.java file shown below:



1. It returns a string with each character one character smaller than the original.
2. Examine the OneTimePad.java shown below:



1. It defines a set of random character replacements for the input string.
2. Uncomment the lines in Tester.java and execute the file as a Java application. You should see:



1. Notice how the client is decoupled from the mechanism of the encryption. It just calls the same method on either implementation.

Congratulations. You have completed this lab.