Proxy

In this lab, you will create a Factory which will return Proxies to remote objects.

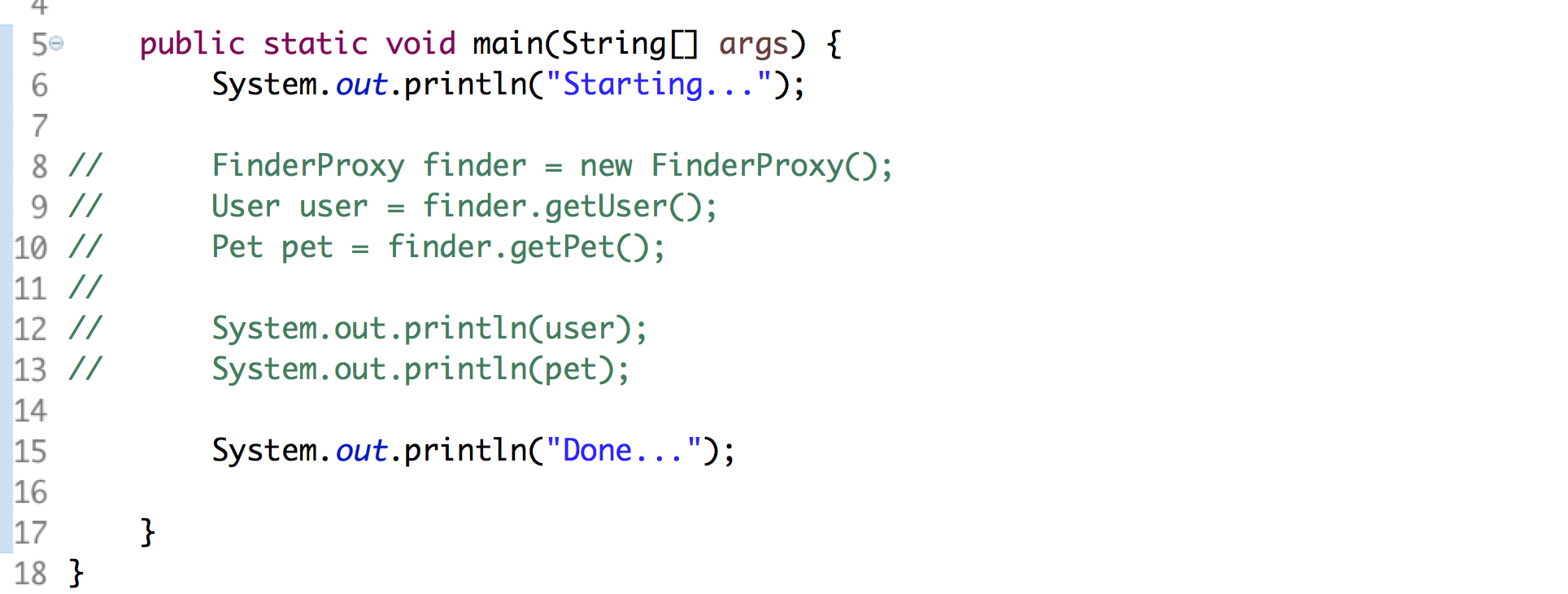
# Objectives

In this lab, you will

* Create a User and a Pet type
* Create RemoteUser and RemotePet classes
* Create FinderProxy as a Factory to return proxies to the remote objects
* Run the app

# Exercise

1. The Proxy design pattern represents interchangeable implementations of the same Interface. Typically, a Factory gets the Proxy object on behalf of a client. Types of proxies include Remote Proxy, Caching Proxy, Authentication Proxy, etc. Very common in SW architecture.
2. In Eclipse, in the exercises workspace, open the package com.paypal.patterns.Proxy to view the project files.
3. Open Tester.java shown below:



1. On lines 8-10, the FinderProxy will do all the work and retrieve the RemoteUser and RemotePet.
2. NOTE: the client does NOT know nor care how the FinderProxy gets / finds the objects. We could replace it with other mechanisms and the client would not care.
3. The User and Pet interfaces are very simple and shown below:



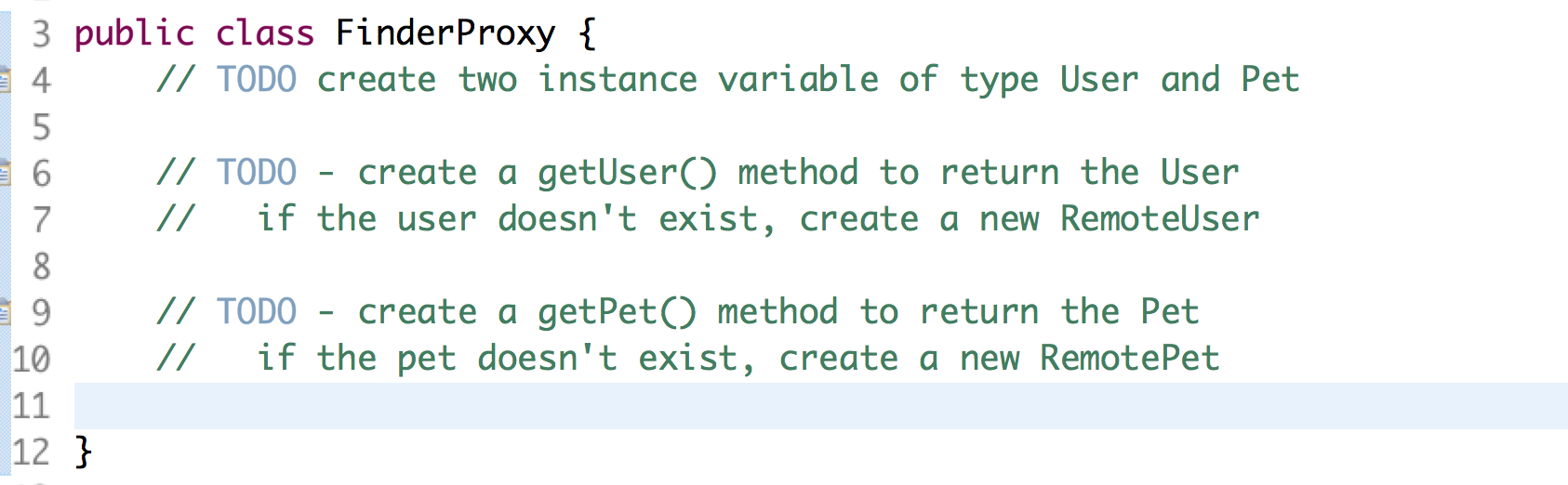


1. The RemoteUser and RemotePet implement basic Java Beans with a single instance variable, name, shown below:

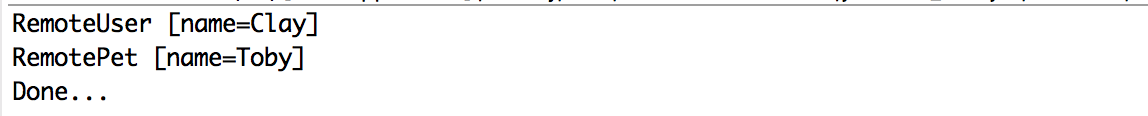




1. The FinderProxy is responsible for getting the remote objects and returning them to the client when requested. In this example, just use the new operation to create instances of the Pet and User. In a real application, the FinderProxy would use the network to find the object.
2. Open FinderProxy.java shown below:



1. Complete the three TODOs.
2. Uncomment the lines from Tester.java and execute it as a Java application. You should see:



Congratulations. You have completed this lab.