Builder Pattern

In this lab, you will create an application using the Builder pattern.

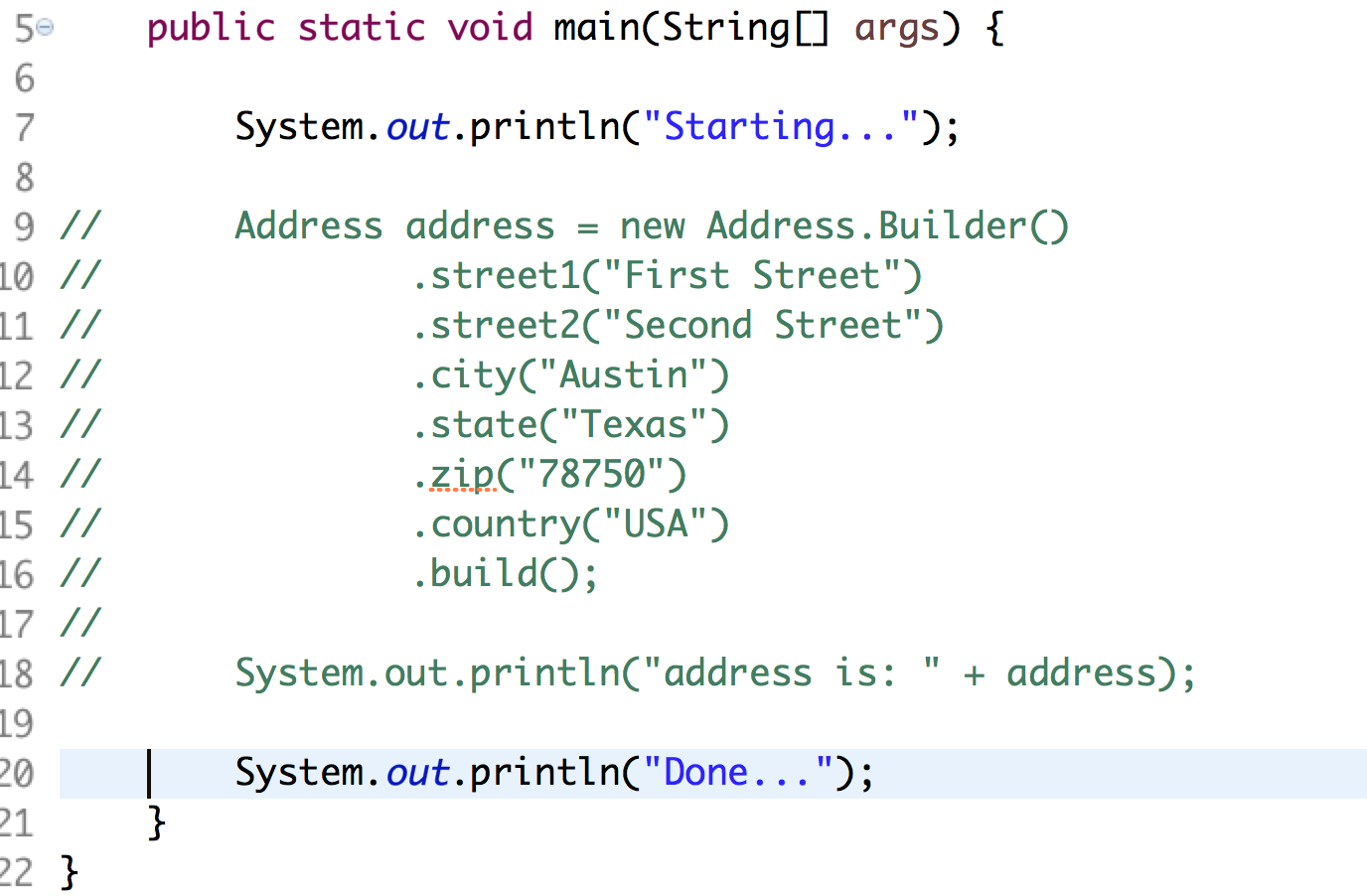
# Objectives

In this lab, you will

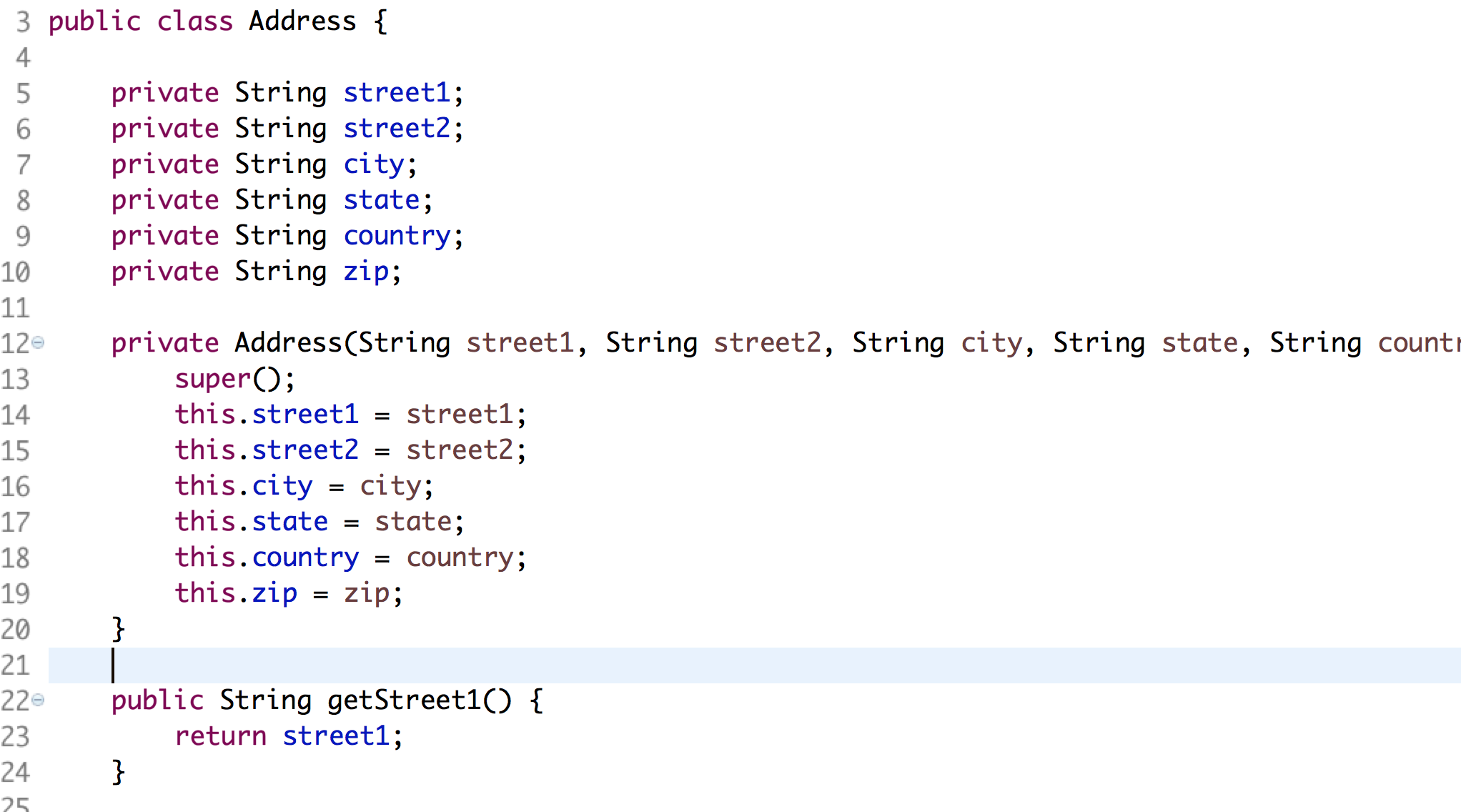
* Create a Builder class inside the Address class
* Use the Address.Builder class to create a new Address object
* Watch it work

# Exercise

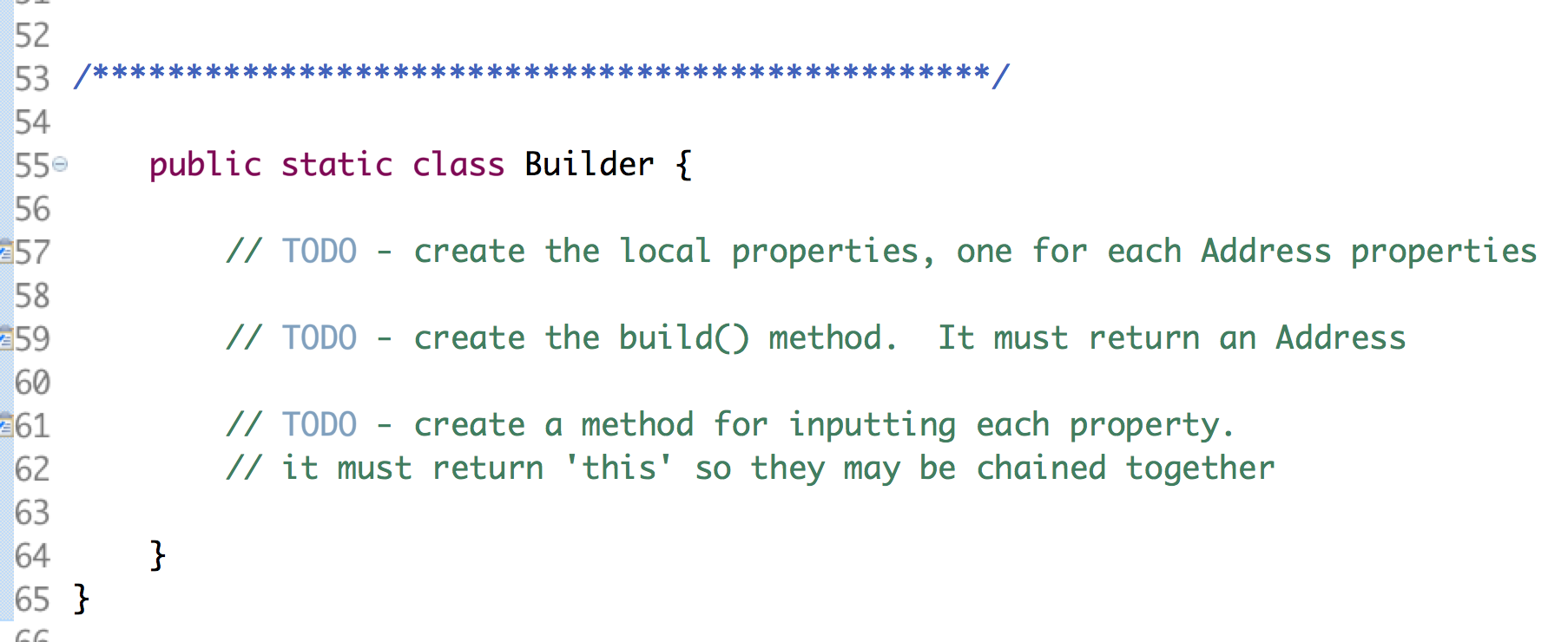
1. The Builder pattern accomplishes two things. First, it generates thread-safe, immutable objects. Second, it makes it easier for the user to create objects with many parameters making the code easier to write and debug.
2. In Eclipse, in the exercises workspace, open the package com.paypal.patterns.Builder to view the project files.
3. Open Tester.java show below:



1. In lines 9-18, the client creates a new Address object using simple methods to describe each parameter. This is very easy to understand and debug. Note the static Builder class only allows instantiation through the Address class. The Address class provides a private constructor, only allowing the Builder to create it.
2. Uncomment the lines so the tester is ready to run.
3. Open Address.java shown below:



1. In line 12, notice the private constructor only allowing the Builder (shown later) to create it. Also notice the constructor contains all the instance properties. As a user, the number of parameters in the constructor would make it very difficult to write and maintain the code.
2. Lines 22-24 define an accessor method for one of the properties. Each property should have its own accessor method. NOTE: there are NO modifiers. This is what makes it thread safe!!!



1. Lines 55-64 defines the class Builder. Notice it is a static class requiring accessing through the containing class, Address.
2. From line 57, create instance properties. There should be one for each of the Address properties.
3. From line 59, create the build() method. It should return an Address object. It should create a new Address() using all the properties in the Builder object.
4. From line 61-62, create an input method for each property. The method must return a Builder object (this), allowing the user to continuously call the builder, appending one method call after another. The last call will be to the build() method which returns the Address object.
5. Execute the Tester as a Java application and watch it work.

Congratulations. You have completed this lab.