Abstract Factory Pattern

In this lab, you will create an Abstract Factory, a factory of factories.

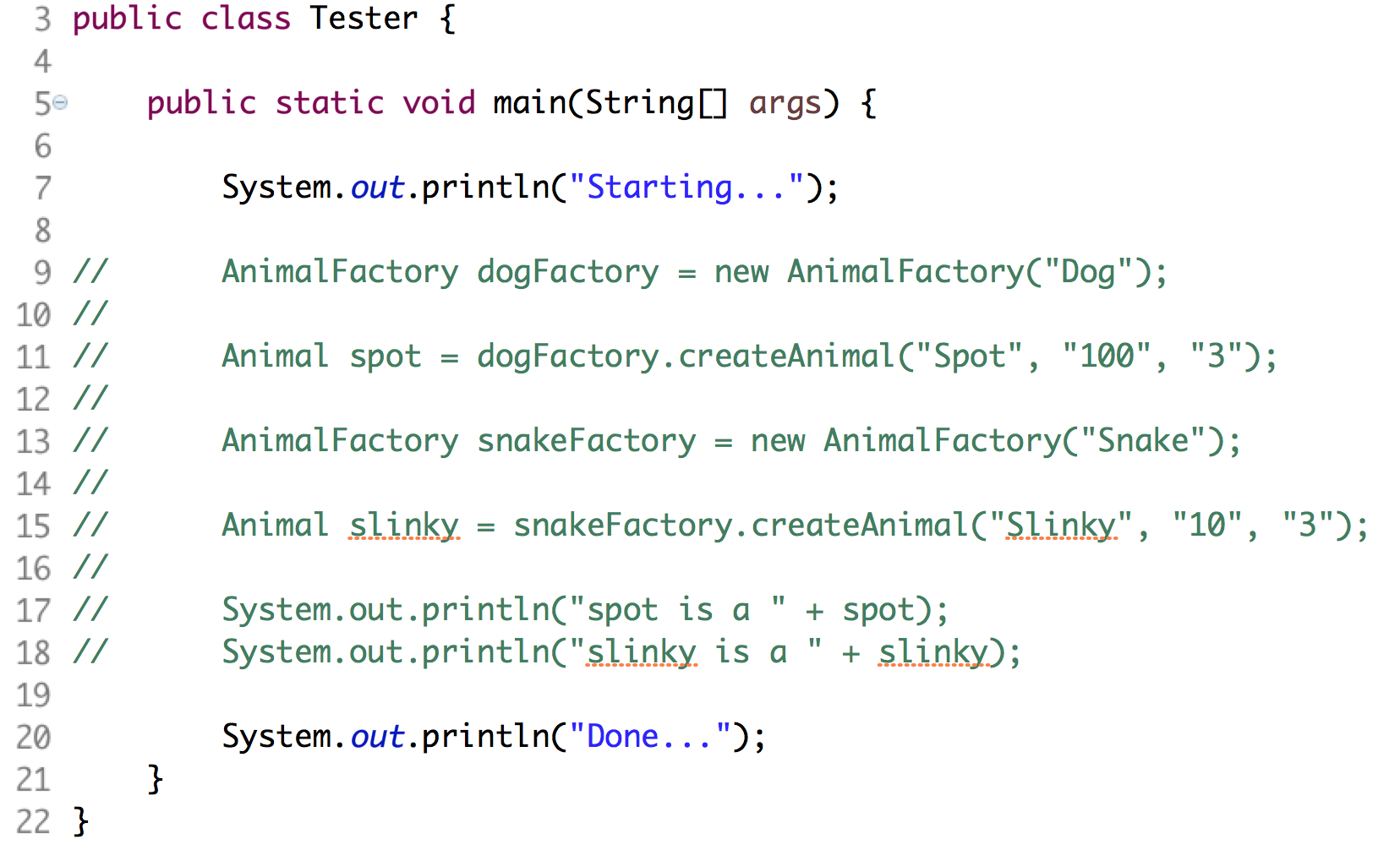
# Objectives

In this lab, you will

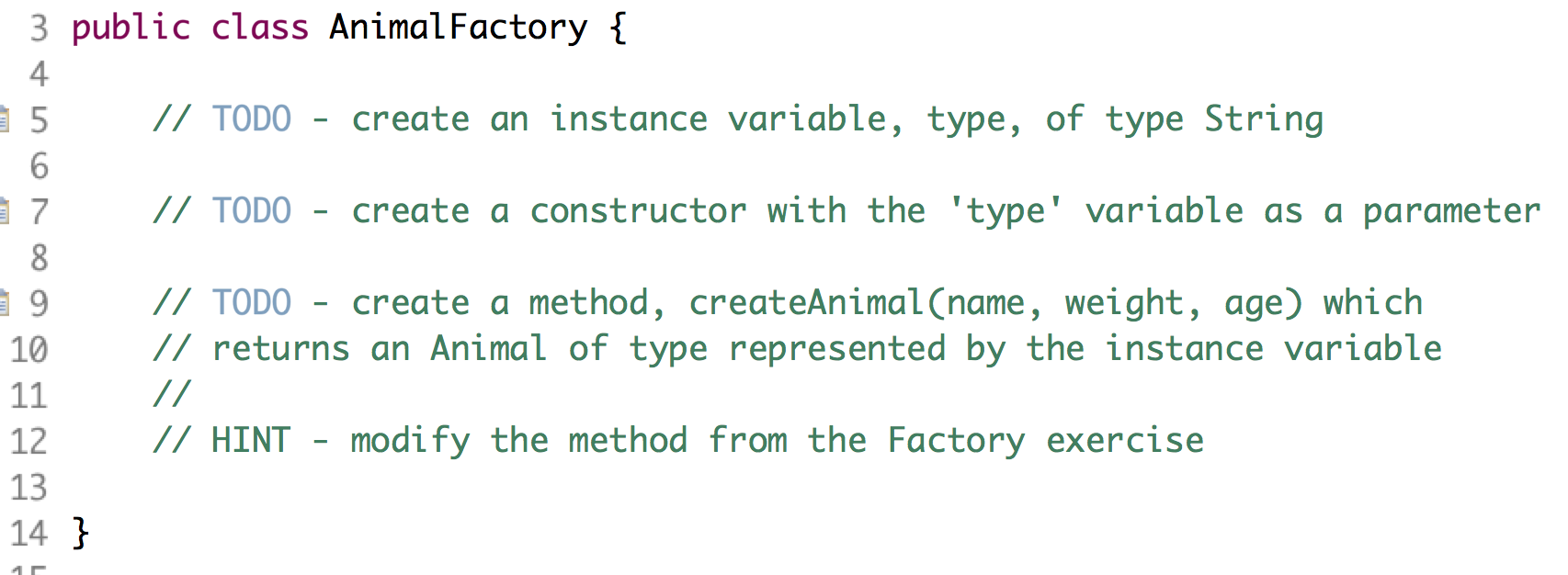
* Create an Abstract Factory of type, AnimalFactory
* Create a Dog Factory and create a Dog
* Create a Snake Factory and create a Snake

# Exercise

1. The Abstract Factory pattern represents a factory of factories. This allows the client code to be independent of which factory the Abstract Factory references. Usually, the Abstract Factory generates the Factory based on run time parameters. For example, it could distinguish between Mac and Windows machines for generating file names. Or, it could read a configuration file to determine which Factory to create based on command line parameters.
2. In Eclipse, in the exercises workspace, open the package com.paypal.patterns.AbstractFactory to view the project files.
3. Open Tester.java as shown below:



1. In lines 9-11 above, the AnimalFactory determines which Factory to create based on the type parameter in the constructor. When we generate the AnimalFactory class, we must adhere to the prototypes indicated in the two lines.
2. Uncomment lines 9 - 18 and ignore the errors until we fix the AnimalFactory class.
3. Open AnimalFactory.java as shown below:



1. From line 5, create an instance variable named type. This represents the type of Animal and was originally passed as the first parameter in the createAnimal() method in the Factory exercise.
2. From line 7, insert the constructor of the class with the type parameter.
3. From line 9, create the method, createAnimal(), using the three parameters: name, weight, and age. NOTE: this method resembles the createAnimal() method from the Factory exercise except the type value comes from the instance variable instead of a method parameter.
4. Run the test program and watch it work…

Congratulations. You have completed this lab.