Factory Pattern

In this lab, you will create a Factory to produce Pets.

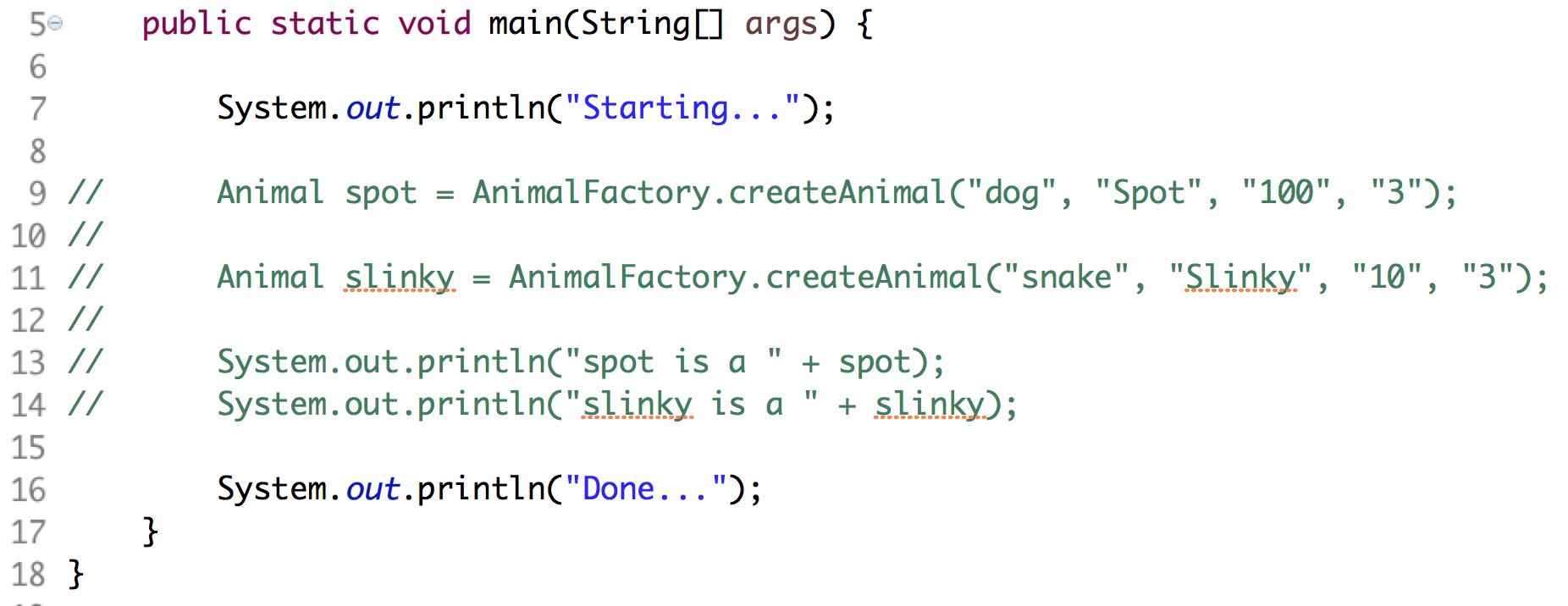
# Objectives

In this lab, you will

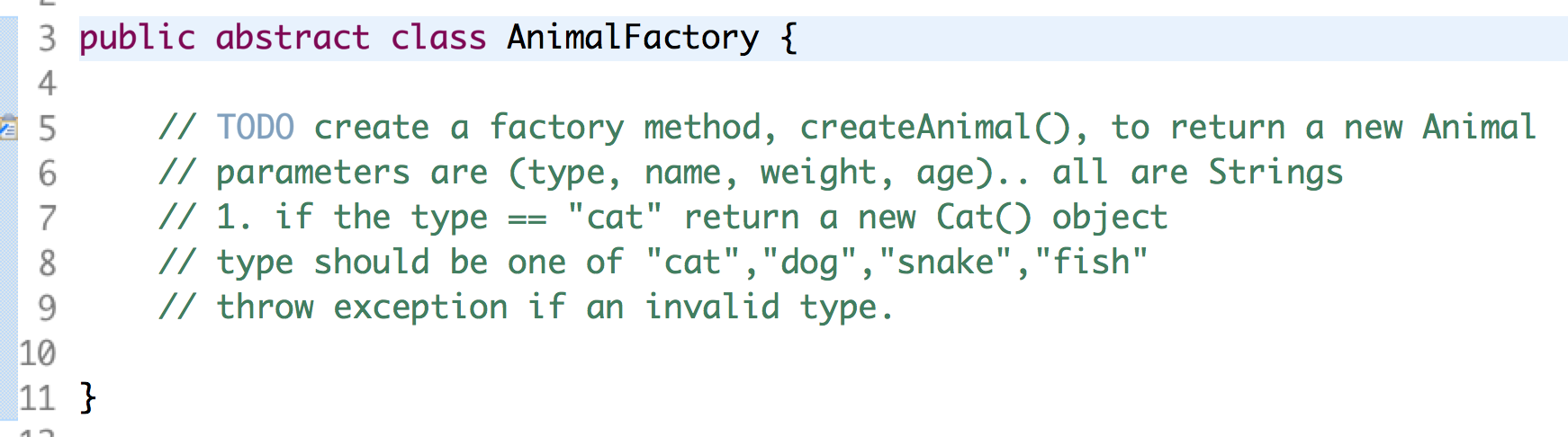
* Create a Pet Factory
* Create several Pets
* Examine the results

# Exercise

1. This program uses the Factory pattern to generate Pets. The Factory pattern allows the client to create objects without prior knowledge of initialization parameters. Otherwise, each client would have to know too much about the internals of the objects. With the Factory pattern, this knowledge is centralized to the Factory.
2. In Eclipse, in the exercises workspace, open the package com.paypal.patterns.Factory to view the project files.
3. Open the file, Tester.java



1. On lines 9 and 11 above, notice the createAnimal() method is a static method. It creates Animals based on the first parameter. If the first parameter is a “dog”, it creates a Dog object. This isolates the client program (Tester) from understanding the internals of HOW to create a Dog object.
2. The AnimalFactory.createAnimal() method must adhere to this signature.
3. Uncomment the lines and notice there are errors in the program until we fix the AnimalFactory.
4. Open AnimalFactory.java as shown below:



1. In the above, since the class is “abstract”, the client cannot issue a new to create an instance of it.
2. On line 5, create a factory method (public and static) named createAnimal() with the four indicated parameters.
3. In the method, use a switch statement or a bunch of if statements to create the appropriate Animal object. Note the Animal objects are defined in the same package.
4. Run the test program, Tester, and watch the results.

Congratulations. You have completed this lab.