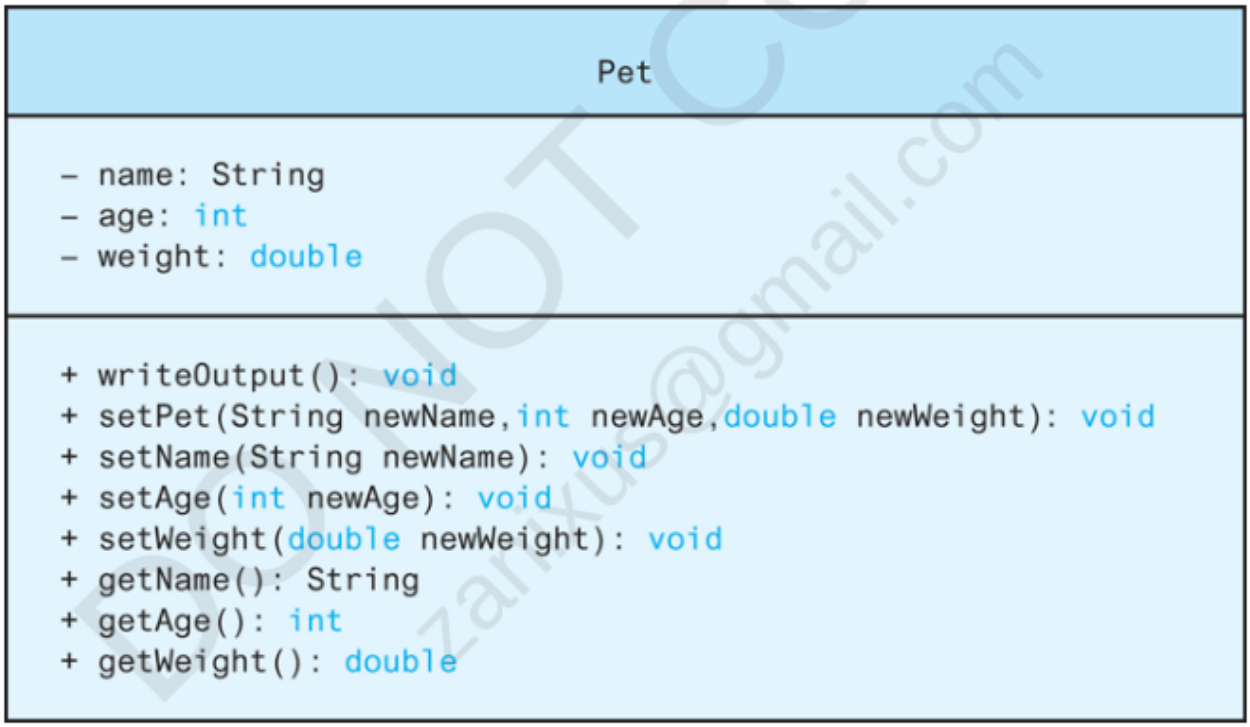
PRINTED BY: zanixus@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Let's consider a simple class to represent our pets. Suppose that we describe a pet by its name, its age, and its weight. We could give basic behaviors to a pet object that simply set or get these three attributes. Now imagine that we draw a class diagram like the one in **Figure 6.1** for a class named `Pet`. Notice the four methods that set different instance variables. One sets all three of the instance variables `name, age`, and `weight`. The other three methods set only one instance variable each. This class diagram does not include constructors, as is typical.

> A class diagram does not include constructors

Figure 6.1 Class Diagram for a Class Pet



One property of constructors that may seem strange at first is that each constructor has the same name as its class. So if the class is named `Species`, its constructors are named `Species`. If the class is named `Pet`, the constructors are named `Pet`.

Constructors often have multiple definitions, each with different numbers or types of parameters, and they sometimes parallel a class's set methods. As an example, **Listing 6.1** contains a definition of our class `Pet` that includes several constructors. Note that the headings of these constructors do not have the word `void`. When you define a constructor, you do not specify any return type, not even `void`.

These constructors look very much like our set methods, which are overloaded.

PRINTED BY: zanixus@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

fact, we have grouped each constructor with its similar set method, just for this example. Unlike some of the set methods, however, the constructors give values to *all* the instance variables, even though they might not have a parameter for each instance variable. If you do not initialize a particular instance variable, Java will do so by giving it a default value. However, it is normal programming practice to explicitly give values to all the instance variables when defining a constructor.

DO NOT COPY
zanixus@gmail.com