

The same holds true for methods: instances of subclasses have all methods defined in both the superclass and the subclass. In general, we can say: because a message post is a post, a message-post object has everything that a post has, and more. And because a photo post is also a post, it has everything that a post has, and more.

Thus, inheritance allows us to create two classes that are quite similar, while avoiding the need to write the identical part twice. Inheritance has a number of other advantages, which we discuss below. First, however, we will take another, more general look at inheritance hierarchies.

## 10.3

## Inheritance hierarchies

### Concept

Classes that are linked through inheritance relationships form an **inheritance hierarchy**.

Inheritance can be used much more generally than shown in the example above. More than two subclasses can inherit from the same superclass, and a subclass can, in turn, be a superclass to other subclasses. The classes then form an *inheritance hierarchy*.

The best-known example of an inheritance hierarchy is probably the classification of species used by biologists. A small part is shown in Figure 10.6. We can see that a poodle is a dog, which is a mammal, which is an animal.

We know some things about poodles—for example, that they are alive, they can bark, they eat meat, and they give birth to live young. On closer inspection, we see that we know some of these things not because they are poodles, but because they are dogs, mammals, or animals. An instance of class **Poodle** (an actual poodle) has all the characteristics of a poodle, a dog, a mammal, and an animal, because a poodle is a dog, which is a mammal, and so on.

The principle is simple: inheritance is an abstraction technique that lets us categorize classes of objects under certain criteria and helps us specify the characteristics of these classes.

**Exercise 10.3** Draw an inheritance hierarchy for the people in your place of study or work. For example, if you are a university student, then your university probably has students (first-year students, second-year students, . . . ), professors, tutors, office personnel, etc.

**Figure 10.6**

An example of an inheritance hierarchy

