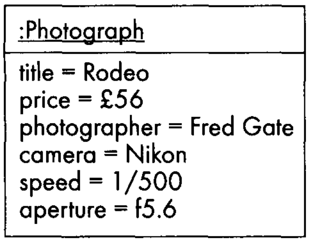

Figure 4.15 Photograph object showing its own and its inherited attributes

When we create a specialized class, it inherits all the attributes, operations and relationships of the parent class. In Figure 4.14, the specialized classes Photograph and Painting inherit the attributes title and price and the operation updatePrice() from the Picture class; they also each have attributes and operations that are relevant only for objects of their specialized class. Notice that the inherited characteristics are not shown in the subclasses; these are the shared features that justify the generalization. This economy of representation simplifies the diagram. However, inherited features do form part of the structure of the inheriting object. A Photograph object would have the attributes: title and price (inherited from Picture) as well as photographer, camera, speed and aperture (see Figure 4.15). It will know about the inherited operation updatePrice() as well as the operation alterContrast().

When we create a set of subclasses we must have some basis for differentiating the subclasses from each other; this is known as the *discriminator.* In the art gallery classes in Figure 4.14 the basis for the differentiation is the type of picture. Usually there will be one subclass for each possible value of the discriminator. As another example, in Figure 4.16 the discriminator is the type of publication; in this case there are three types of publication, therefore there are three subclasses.

It is easy to misuse inheritance; it is a useful technique only if used correctly. In the past programmers have been tempted to introduce a class from another system and specialize it simply so that they can use one if its methods.[4] The reused class might have nothing in common with the new system except for a useful algorithm. This can be confusing for those reading the code; their expectations of the inheritance relationship are confounded. Classes should not be linked by inheritance unless there is an *is-a* or *is-a-kind-of* relationship between them: in the art gallery example photograph and painting are both *a-kind-of* picture.

4. *A method is the way that an operation is implemented; this is discussed later in this chapter.*