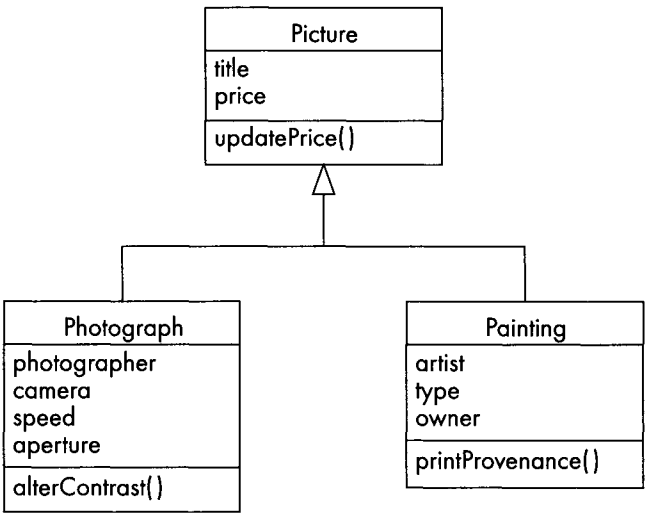

Figure 4.27 *Picture, Painting and Photograph classes restructured to conform to the principle of substitutability*

Composition

The UML has a stronger form of aggregation known as composition (it is also sometimes referred to as aggregation containment). Composition is like aggregation in that it models a whole–part relationship between objects, but unlike aggregation in that it has a very precise meaning. In a composition relationship:

- The whole object has exclusive ownership of its parts, i.e. the part object can only participate in one aggregation
- Part objects, therefore, have no separate existence from the whole
- The parts live and die with the whole, i.e.
  - the whole creates its parts and
  - when the whole is deleted, its parts are deleted (a cascading delete).

Part objects are hidden (contained) within the whole in that they are not visible to the rest of the program. In the same way that operations forming the interface of an object encapsulate and hide the object's data, the interface of the whole object encapsulates and hides its part objects. The rest of the program can communicate only with the whole; any communication with the parts is done by the whole object. The notation for a composition relationship is the same as for aggregation, but with a black diamond instead of a white one. This is shown in Figure 4.28, which illustrates a composition relationship between a Robot and its parts.

If we want to get the Robot to pick up a glass, we send the message to the Robot object, not to its hand.