Notations for Figures

We have used UML-*like* class diagrams throughout the book. In this appendix, we succinctly describe the notations we have used in this book for such class diagrams with the help of a figure (refer Figure C.1).

With respect to the Figure C.1, following notations are used in this book:

- A rectangle represents an entity such as an interface, an abstract class, a concrete class, or an enumeration. Some of the rectangles may have three partitions:
  - The first partition shows the name of the type. If the type is an interface or abstract class, then keywords "≪interface≫" or "≪abstract - class≫" appear before the name of the type.
  - The second partition lists the used data members along with the access specifier as shown in `Image` class, i.e., "`-imageType: ImageType`." The "–" and "+" symbol prefixes denote private and public members, respectively.
  - The third partition lists all the supported methods of the type along with parameters and access specifier; for instance, the `Image` class supports a

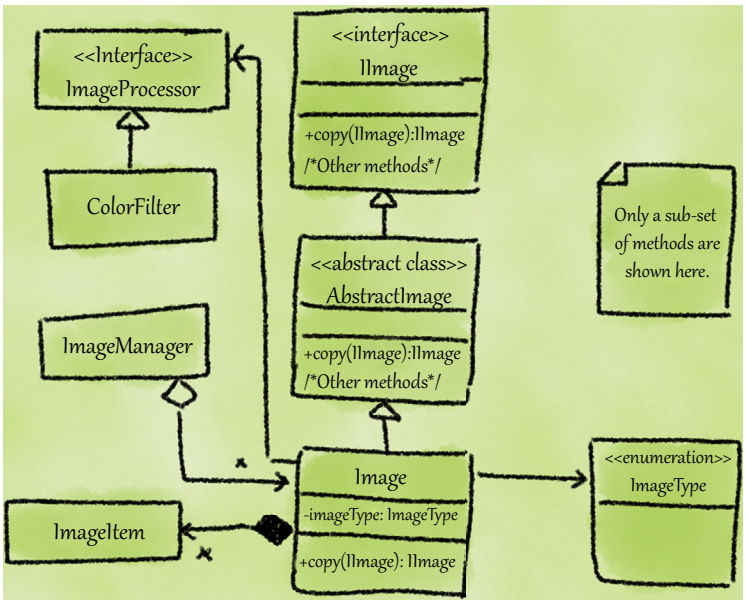

FIGURE C.1

An example to illustrate notations used in this book for UML-like class diagrams.

public method `copy()` that accepts an input parameter of type `Image` and returns an `Image` instance, which is shown by "`+copy(IImage): IImage`."

- A solid line with a hollow triangle at one end denotes an inheritance relationship between two types; for instance, in this figure, `Image` extends `AbstractImage`.
- A solid directed arrow denotes an association between two types. In this figure, an `Image` has a reference to `ImageProcessor`.
- A line with a filled diamond at one end and an arrow at the other end shows a composition relationship. In this figure, `Image` is composed of `ImageItem` objects. The symbol * near the arrow shows multiplicity, i.e., `Image` contains zero or more `ImageItem` objects.
- A line with a hollow diamond at one end and an arrow at the other end shows an aggregation relationship. In this figure, `ImageManager` is an aggregation of `Image` objects.
- The keyword "≪enumeration≫" along with the name of the type represents an enumeration type; in this figure `ImageType` is an enumeration type.
- A "dog-eared" rectangle is used for conveying a note or comment to the reader (such as the note shown toward the right top corner of this figure).