CASE STUDY Processing Geometric Figures

Problem We would like to process some standard geometric shapes. Each `figure` object will be one of three standard shapes (rectangle, circle, and right triangle). We would like to be able to do standard computations, such as finding the area and perimeter, for any of these shapes.

Analysis For each of the geometric shapes we can process, we need a class that represents the shape and knows how to perform the standard computations on it (i.e., find its area and perimeter). These classes will be `Rectangle`, `Circle`, and `RtTriangle`. To ensure that these shape classes all define the required computational methods (finding area and perimeter), we will make them abstract methods in the base class for the shape hierarchy. If a shape class does not have the required methods, we will get a syntax error when we attempt to compile it.

Figure 1.10 shows the class hierarchy. We used abstract class `Shape` as the base class of the hierarchy. We didn't consider using an actual class because there are no actual objects of the base class type. The single data field `shapeName` stores the kind of shape object as a `String`.

Design We will discuss the design of the `Rectangle` class here. The design of the other classes is similar and is left as an exercise. Table 1.7 shows class `Rectangle`. Class `Rectangle` has data fields `width` and `height`. It has methods to compute area and perimeter, a method to read in the attributes of a rectangular object (`readShapeData`), and a `toString` method.

FIGURE 1.10
Abstract Class Shape and Its Three Actual Subclasses

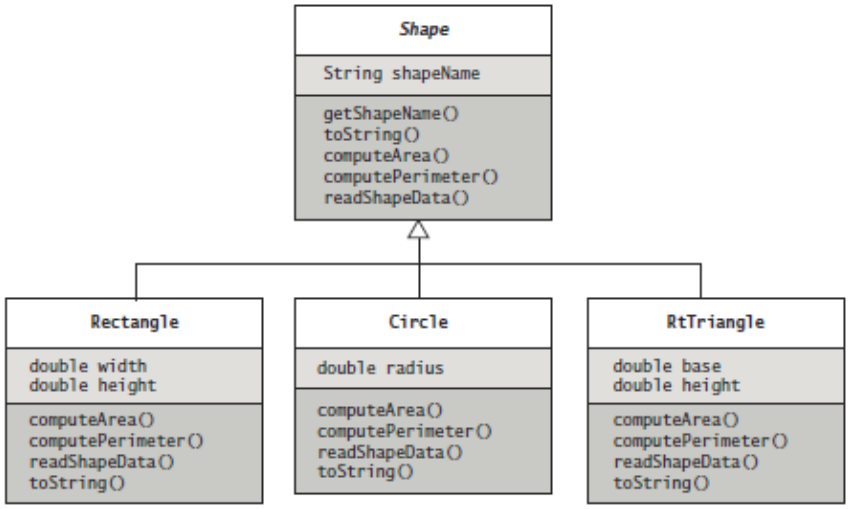

TABLE 1.7
Class `Rectangle`

Data Field	Attribute
`double width`	Width of a rectangle
`double height`	Height of a rectangle
Method	**Behavior**
`double computeArea()`	Computes the rectangle area (width × height)
`double computePerimeter()`	Computes the rectangle perimeter (2 × width + 2 × height)
`void readShapeData()`	Reads the width and height
`String toString()`	Returns a string representing the state

Implement the listing 1.6 (Abstract class Shape) and listing 1.7 (Rectangle class), use listing 1.8 (provided for downloading) to test your Rectangle class.

Please submit all .java source codes into the dropbox.