Ch. 8 CSE3130 - Object Oriented Programming 2 - Skillbuilders

Zephram Gilson

`Rectangle`, rect, & ComparableArea

Part 1:

1. The `Rectangle` class includes two constructors:
    - A default constructor (when no parameters are specified) that sets the length and width to `1`.
    - A parameterized constructor that accepts specific values for length and width.
2. The class includes getter and setter methods for both length and width, and methods to calculate the `area` and `perimeter`.
3. The client code `rect.java` tests both constructors, displays the rectangle's dimensions, and calculates the area and perimeter. It also tests the setter methods to update the dimensions.

Ensures that rectangles can be created and/or manipulated with or without initial dimensions.

Part 2:

`displayAreaFormula()` method: static method which prints the formula for calculating the area of a rectangle, which is `Area = length * width`.

- Since it's a class method, it doesn't operate on specific instances and can be called using the class name (`Rectangle.displayAreaFormula()`).

Client code now includes a call to `Rectangle.displayAreaFormula()` to display the area formula, demonstrating the use of the static method.

Part 3:

`equals()` method: checks if two rectangles have the same length and width. If both the length and width of the two objects match, the method returns `true`; otherwise, it returns `false`.
`toString()` method: provides a meaningful string representation of the rectangle, including its length and width.

Client Code now includes tests for the `equals()` method by comparing two rectangles (ex. `rect2` and `rect3` have the same dimensions, while `rect4` has different dimensions).

Part 4:

`Comparable<Rectangle>:` The `Rectangle` class now implements the `Comparable` interface, allowing rectangles to be compared based on their area.
`compareTo()` method: compares the current rectangle's area to another rectangle's area. It returns:

- A negative number if the current rectangle has a smaller area.
- Zero if both rectangles have the same area.
- A positive number if the current rectangle has a larger area.

The **client code** now includes a test for sorting an array of rectangles using `Arrays.sort()`. The rectangles will be sorted by their area.

Part 5:

`ComparableArea` interface: This new interface defines the method `compareToArea()` which compares the areas of two rectangles.
`Rectangle` class: Now implements both `Comparable<Rectangle>` and `ComparableArea.` The `compareTo()` method compares areas for sorting, and `compareToArea()` provides a more flexible way of comparing specific rectangle objects by their area.

Client code now includes a test for `compareToArea(),` comparing two rectangles by their area and outputting the result.