2.21 A vending machine that dispenses bottled water has seven features in its interface. See Fig. 2.26.

The features consist of three buttons (S, M, L) for selecting the desired bottle size (small, medium, or large), a slot to insert coins, an output tray in which the bottle drops, a hexagonal button (R) for requesting change or coin return, and a red light that flashes once (lights up momentarily) if the machine encounters a problem. To purchase a small bottle, for example, insert 35 cents and press the S, and a small bottle will drop into the output tray. Express these features as methods, complete with signatures and returns.

2.22 Let *selectSmall()* be the method corresponding to pressing the S button of the vending machine in Exercise 2.22. Write its pre and post.

eCheck

Check02A (TS = 16)

The UML diagram of the Math utility class of `java.lang` contains the following two entries in the method compartment:

```
rint(double) : double
round(double) : long
```

Write the app Check02A that explores the difference between these two methods by reading a double from the user and outputting the result of invoking each. Here is a sample run of the sought app:

```
Enter any double to test: 12.43
Method Math.rint returns: 12.0
While Math.round returns: 12
```

Here is another sample run:

```
Enter any double to test: 262.5
Method Math.rint returns: 262.0
While Math.round returns: 263
```

Implement your app, test it, and then eCheck it. Keep in mind that your app must work correctly for any input, not just the ones shown above. Note also that your app must adhere to the exact output layout shown above. No number formatting whatsoever is needed; output the returns of the two methods as-is.

There is a subtle difference between the two methods (in addition to the obvious difference in return type): the first method returns the closest integral double to the passed parameter (and if two such values are equally close, the even one breaks the tie). The second adds 0.5 to the parameter and returns the floor of the sum as long. This algorithmic difference may lead to an observable difference as seen in the second sample run.