

2. The class `SingleTable` represents a table at a restaurant.

```
public class SingleTable
{
    /** Returns the number of seats at this table. The value is always greater than or equal to 4. */
    public int getNumSeats()
    { /* implementation not shown */ }

    /** Returns the height of this table in centimeters. */
    public int getHeight()
    { /* implementation not shown */ }

    /** Returns the quality of the view from this table. */
    public double getViewQuality()
    { /* implementation not shown */ }

    /** Sets the quality of the view from this table to value. */
    public void setViewQuality(double value)
    { /* implementation not shown */ }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

At the restaurant, customers can sit at tables that are composed of two single tables pushed together. You will write a class `CombinedTable` to represent the result of combining two `SingleTable` objects, based on the following rules and the examples in the chart that follows.

- A `CombinedTable` can seat a number of customers that is two fewer than the total number of seats in its two `SingleTable` objects (to account for seats lost when the tables are pushed together).
- A `CombinedTable` has a desirability that depends on the views and heights of the two single tables. If the two single tables of a `CombinedTable` object are the same height, the desirability of the `CombinedTable` object is the average of the view qualities of the two single tables.
- If the two single tables of a `CombinedTable` object are not the same height, the desirability of the `CombinedTable` object is 10 units less than the average of the view qualities of the two single tables.

Assume `SingleTable` objects `t1`, `t2`, and `t3` have been created as follows.

- `SingleTable t1` has 4 seats, a view quality of 60.0, and a height of 74 centimeters.
- `SingleTable t2` has 8 seats, a view quality of 70.0, and a height of 74 centimeters.
- `SingleTable t3` has 12 seats, a view quality of 75.0, and a height of 76 centimeters.

The chart contains a sample code execution sequence and the corresponding results.

Statement	Value Returned (blank if no value)	Class Specification
<code>CombinedTable c1 = new CombinedTable(t1, t2);</code>		A <code>CombinedTable</code> is composed of two <code>SingleTable</code> objects.
<code>c1.canSeat(9);</code>	true	Since its two single tables have a total of 12 seats, <code>c1</code> can seat 10 or fewer people.
<code>c1.canSeat(11);</code>	false	<code>c1</code> cannot seat 11 people.
<code>c1.getDesirability();</code>	65.0	Because <code>c1</code> 's two single tables are the same height, its desirability is the average of 60.0 and 70.0.
<code>CombinedTable c2 = new CombinedTable(t2, t3);</code>		A <code>CombinedTable</code> is composed of two <code>SingleTable</code> objects.
<code>c2.canSeat(18);</code>	true	Since its two single tables have a total of 20 seats, <code>c2</code> can seat 18 or fewer people.
<code>c2.getDesirability();</code>	62.5	Because <code>c2</code> 's two single tables are not the same height, its desirability is 10 units less than the average of 70.0 and 75.0.
<code>t2.setViewQuality(80);</code>		Changing the view quality of one of the tables that makes up <code>c2</code> changes the desirability of <code>c2</code> , as illustrated in the next line of the chart. Since <code>setViewQuality</code> is a <code>SingleTable</code> method, you do not need to write it.
<code>c2.getDesirability();</code>	67.5	Because the view quality of <code>t2</code> changed, the desirability of <code>c2</code> has also changed.

The last line of the chart illustrates that when the characteristics of a `SingleTable` change, so do those of the `CombinedTable` that contains it.

Write the complete `CombinedTable` class. Your implementation must meet all specifications and conform to the examples shown in the preceding chart.

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

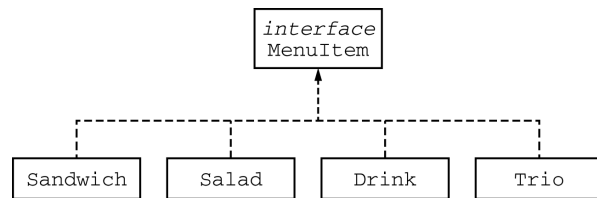
4. The menu at a lunch counter includes a variety of sandwiches, salads, and drinks. The menu also allows a customer to create a "trio," which consists of three menu items: a sandwich, a salad, and a drink. The price of the trio is the sum of the two highest-priced menu items in the trio; one item with the lowest price is free.

Each menu item has a name and a price. The four types of menu items are represented by the four classes Sandwich, Salad, Drink, and Trio. All four classes implement the following MenuItem interface.

```
public interface MenuItem
{
    /** @return the name of the menu item */
    String getName();

    /** @return the price of the menu item */
    double getPrice();
}
```

The following diagram shows the relationship between the MenuItem interface and the Sandwich, Salad, Drink, and Trio classes.



For example, assume that the menu includes the following items. The objects listed under each heading are instances of the class indicated by the heading.

Sandwich	Salad	Drink
"Cheeseburger" 2.75	"Spinach Salad" 1.25	"Orange Soda" 1.25
"Club Sandwich" 2.75	"Coleslaw" 1.25	"Cappuccino" 3.50

Question 4 continues on page 13

The menu allows customers to create Trio menu items, each of which includes a sandwich, a salad, and a drink. The name of the Trio consists of the names of the sandwich, salad, and drink, in that order, each separated by "/" and followed by a space and then "Trio". The price of the Trio is the sum of the two highest-priced items in the Trio; one item with the lowest price is free.

A trio consisting of a cheeseburger, spinach salad, and an orange soda would have the name "Cheeseburger/Spinach Salad/Orange Soda Trio" and a price of \$4.00 (the two highest prices are \$2.75 and \$1.25). Similarly, a trio consisting of a club sandwich, coleslaw, and a cappuccino would have the name "Club Sandwich/Coleslaw/Cappuccino Trio" and a price of \$6.25 (the two highest prices are \$2.75 and \$3.50).

Write the Trio class that implements the MenuItem interface. Your implementation must include a constructor that takes three parameters representing a sandwich, salad, and drink. The following code segment should have the indicated behavior.

```
Sandwich sandwich;
Salad salad;
Drink drink;
/* Code that initializes sandwich, salad, and drink */

Trio trio = new Trio(sandwich, salad, drink); // Compiles without error

Trio trio1 = new Trio(salad, sandwich, drink); // Compile-time error
Trio trio2 = new Trio(sandwich, salad, salad); // Compile-time error
```

WRITE YOUR SOLUTION ON THE NEXT PAGE.

Write the complete `Trio` class below.

STOP
END OF EXAM