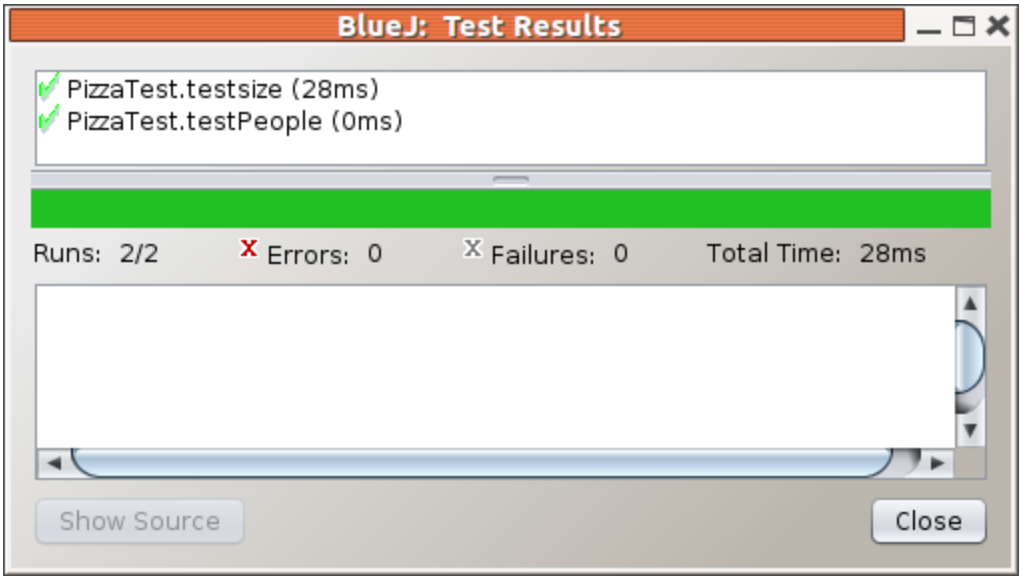
```
import java.io.*;
import java.util.Scanner;
/**
 *   PizzaCalculator is a simple calculator for computing how the
 *   price of a pizza is divided among people sharing it.
 *
 * @author John Dalbey
 * CSc 103-03
 * @version 2014.4.10
 **/
public class PizzaCalculator
{
    final static String kPrompt = "Enter diameter, price (xx.xx), and number of consumers: ";
    /**
     * Run the console user interface, obtaining data from standard input
     * and displaying results on standard output.
     */
    public void run()
    {
        Scanner console = new Scanner(System.in);

        System.out.println(kPrompt);
        // Read pizzas until End of File
        while (console.hasNext())
        {
            // Enter pizza attributes
            int diameter = console.nextInt();
            String price = console.next();
            int people = console.nextInt();
            // Make the pizza
            Pizza pie = new Pizza(diameter, price, people);
            // Display results
            System.out.printf("Portion size: %2.2f sq. inches\n", pie.getPortionSize());
            System.out.println("Each person owes: $" + pie.getPricePerPerson());
            System.out.println(kPrompt);
        }
    }
    /**
     * Entry point for the application.
     * @param args ignored
     */
    public static void main(String[] args)
    {
        new PizzaCalculator().run();
    }

}
```

```
/**
 * Pizza represents a traditional circular Italian
 * bread dinner dish, consumed by several people.
 * The portion size and price per person can be computed.
 * @author John Dalbey
 * @version 2014.4.10
 */
public class Pizza
{
    private int diameter;           /* Diameter of pizza in inches */
    private int price;              /* Price of pizza in dollars and cents */
    private int people;             /* Number of people eating */
    private double portionSize;     /* The size of each person's portion of the pizza */

    /** Construct a pizza of a given diameter and price,
     * to be eaten by the specified number of people.
     * @param diameter how big the pizza is (inches)  (assumed non-negative)
     * @param price how much it costs (in dollars and cents) (assumed non-negative)
     * @param people number of consumers (assumed > 0)
     */
    public Pizza(int diameter, String price, int people)
    {
        this.diameter = diameter;
        this.price = getPennies(price);
        this.people = people;
        this.portionSize = calcArea() / people;
    }

    /**
     * Convert dollars and cents to pennies. Assumes amount is non-negative.
     * @param amount String representation of dollars and cents, e.g., "5.20"
     * @return the whole number of cents equivalent to amount.
     */
    public static int getPennies(String amount)
    {
        int length = amount.trim().length();
        // convert the cents portion
        int cents = Integer.parseInt(amount.substring(length-2));
        // convert the dollars portion
        String dollars = amount.substring(0,length-3);
        cents += Integer.parseInt(dollars) * 100;
        return cents;
    }

    /**
     * Change the number of people eating this pizza.
     * Assumes people > 0
     * @param numPeople the number of consumers
     */
    public void setPeopleCount(int numPeople)
    {
        this.people = numPeople;
        this.portionSize = calcArea() / numPeople;
    }

    /* Compute area of pizza.
     * @return the surface area (in square inches)
     */
    private double calcArea()
    {
        double radius = diameter / 2.0;
        return radius * radius * Math.PI;
    }

    /**
     * Compute the portion each person receives.
     * @return fractional amount of pizza per person (square inches)
     */
    public double getPortionSize()
    {
        return portionSize;
    }
```

```
    /**
     * Compute the amount each person pays, including tax and tip.
     * @return a string representation in dollars and cents for the
     * amount each person pays.
     */
    public String getPricePerPerson()
    {
        final double taxRate = 0.0725;
        final double tipPercent = 0.20;
        double gratuity = price * tipPercent;  // compute tip
        // total includes tax and tip
        double totalPrice = price + price * taxRate + gratuity;
        /* divide by number of people and adjust for float precision errors
         * then convert pennies to dollars & cents */
        double unitPrice = Math.ceil(totalPrice / people) / 100;
        return String.format("%6.2f", unitPrice).trim();
    }
}
```

```
/**
 * The test class PizzaTest.
 *
 * @author  jdalbey
 * @version 0.1
 */
public class PizzaTest extends junit.framework.TestCase
{

    public void testsize()
    {
        Pizza pizza1 = new Pizza(10, "7.50", 3);
        assertEquals(26.1, pizza1.getPortionSize(), 0.1);
        assertEquals("3.19", pizza1.getPricePerPerson());
    }
    public void testPeople()
    {
        Pizza pizza1 = new Pizza(10, "7.50", 3);
        pizza1.setPeopleCount(2);
        assertEquals(39.2, pizza1.getPortionSize(), 0.1);
        assertEquals("4.78", pizza1.getPricePerPerson());
    }

}
```

```
$ javac PizzaCalculator.java
$ java -cp /usr/local/bluej253/bluej/lib/junit.jar:. junit.textui.TestRunner PizzaTest
..
Time: 0.017

OK (2 tests)

$ cat systest_consoledata.txt
16
14.90
3
12
11.50
2

$ java PizzaCalculator < systest_consoledata.txt
Enter diameter, price (xx.xx), and number of consumers:
Portion size: 67.02 sq. inches
Each person owes: $6.33
Enter diameter, price (xx.xx), and number of consumers:
Portion size: 56.55 sq. inches
Each person owes: $7.32
Enter diameter, price (xx.xx), and number of consumers:
$
```



Note a screenshot is not necessary if you provided the output from the text-based TestRunner, as shown on the previous page.