The printing of the error message follows exactly the same pattern as we saw for printing the price of tickets in the **printTicket** method; it is just a little more verbose.

```
System.out.println("You must insert at least: " +
                   (price - balance) + " more cents.");
```

The single actual parameter to the **println** method consists of a concatenation of three elements: two string literals on either side of a numeric value. In this case, the numeric value is a subtraction that has been placed in parentheses to indicate that it is the resulting value we wish to concatenate with the two strings.

Exercise 2.50 In this version of **printTicket**, we also do something slightly different with the **total** and **balance** fields. Compare the implementation of the method in Code 2.1 with that in Code 2.8 to see whether you can tell what those differences are. Then check your understanding by experimenting within BlueJ.

Exercise 2.51 Is it possible to remove the else part of the if-statement in the **printTicket** method (i.e., remove the word **else** and the block attached to it)? Try doing this and seeing if the code still compiles. What happens now if you try to print a ticket without inserting any money?

The **printTicket** method reduces the value of **balance** by the value of **price**. As a consequence, if a customer inserts more money than the price of the ticket, then some money will be left in the balance that could be used toward the price of a second ticket. Alternatively, the customer could ask to be refunded the remaining balance, and that is what the **refundBalance** method does, as we shall see in section 2.16.

2.15 Scope highlighting

You will have noticed by now that the BlueJ editor displays source code with some additional decoration: colored boxes around some elements, such as methods and if-statements (see, for example, Code 2.8).

These colored annotations are known as *scope highlighting*, and they help clarify logical units of your program. A *scope* (also called a *block*) is a unit of code usually indicated by a pair of curly brackets. The whole body of a class is a scope, as is the body of each method and the *if* and *else* parts of an if-statement.

As you can see, scopes are often nested: the if-statement is inside a method, which is inside a class. BlueJ helps by distinguishing different kinds of scopes with different colors.

One of the most common errors in the code of beginning programmers is getting the curly brackets wrong—either by having them in the wrong place, or by having a bracket missing altogether. Two things can greatly help in avoiding this kind of error:

- Pay attention to proper indentation of your code. Every time a new scope starts (after an open curly bracket), indent the following code one level more. Closing the scope brings