

2.18 (GUI) Getting Input from Input Dialogs

You can obtain input from the console. Alternatively, you may obtain input from an input dialog box by invoking the `JOptionPane.showInputDialog` method, as shown in Figure 2.5.

`JOptionPane` class



FIGURE 2.5 The input dialog box enables the user to enter a string.

When this method is executed, a dialog is displayed to enable you to enter an input value. After entering a string, click *OK* to accept the input and dismiss the dialog box. The input is returned from the method as a string.

There are several ways to use the `showInputDialog` method. For the time being, you need to know only two ways to invoke it.

One is to use a statement like this one:

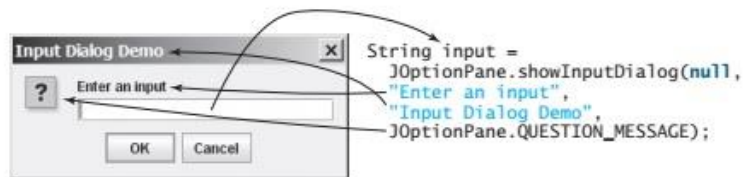
```
JOptionPane.showInputDialog(x);
```

where `x` is a string for the prompting message.

The other is to use a statement such as the following:

```
String string = JOptionPane.showInputDialog(null, x,  
y, JOptionPane.QUESTION_MESSAGE);
```

where `x` is a string for the prompting message and `y` is a string for the title of the input dialog box, as shown in the example below.



`showInputDialog` method

2.18.1 Converting Strings to Numbers

The input returned from the input dialog box is a string. If you enter a numeric value such as `123`, it returns `"123"`. You have to convert a string into a number to obtain the input as a number.

`Integer.parseInt` method

To convert a string into an `int` value, use the `parseInt` method in the `Integer` class, as follows:

```
int intValue = Integer.parseInt(intString);
```

where `intString` is a numeric string such as `"123"`.

`Double.parseDouble` method

To convert a string into a `double` value, use the `parseDouble` method in the `Double` class, as follows:

```
double doubleValue = Double.parseDouble(doubleString);
```

where `doubleString` is a numeric string such as `"123.45"`.

The `Integer` and `Double` classes are both included in the `java.lang` package, and thus they are automatically imported.

2.18.2 Using Input Dialog Boxes

Listing 2.8, `ComputeLoan.java`, reads input from the console. Alternatively, you can use input dialog boxes.

Listing 2.11 gives the complete program. Figure 2.6 shows a sample run of the program.

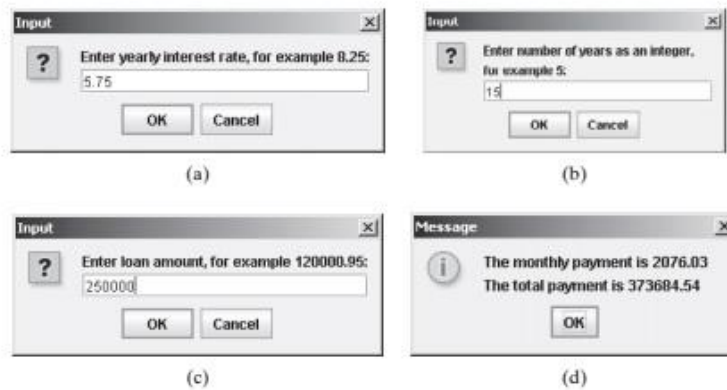


FIGURE 2.6 The program accepts the annual interest rate (a), number of years (b), and loan amount (c), then displays the monthly payment and total payment (d).

LISTING 2.11 `ComputeLoanUsingInputDialog.java`

```
1 import javax.swing.JOptionPane;
2
3 public class ComputeLoanUsingInputDialog {
4     public static void main(String[] args) {
5         // Enter yearly interest rate
6         String annualInterestRateString = JOptionPane.showInputDialog(
7             "Enter yearly interest rate, for example 8.25:");
8
9         // Convert string to double
10        double annualInterestRate =
11            Double.parseDouble(annualInterestRateString);
12
```

enter interest rate

convert string to double

```

13 // Obtain monthly interest rate
14 double monthlyInterestRate = annualInterestRate / 1200;
15
16 // Enter number of years
17 String numberOfYearsString = JOptionPane.showInputDialog(
18     "Enter number of years as an integer, \nfor example 5:");
19
20 // Convert string to int
21 int numberOfYears = Integer.parseInt(numberOfYearsString);
22
23 // Enter loan amount
24 String loanString = JOptionPane.showInputDialog(
25     "Enter loan amount, for example 120000.95:");
26
27 // Convert string to double
28 double loanAmount = Double.parseDouble(loanString);
29
30 // Calculate payment
31 double monthlyPayment = loanAmount * monthlyInterestRate / (1
32     - 1 / Math.pow(1 + monthlyInterestRate, numberOfYears * 12));
33 double totalPayment = monthlyPayment * numberOfYears * 12;
34
35 // Format to keep two digits after the decimal point
36 monthlyPayment = (int)(monthlyPayment * 100) / 100.0;
37 totalPayment = (int)(totalPayment * 100) / 100.0;
38
39 // Display results
40 String output = "The monthly payment is " + monthlyPayment +
41     "\nThe total payment is " + totalPayment;
42 JOptionPane.showMessageDialog(null, output);
43 }
44 }

```

monthlyPayment

totalPayment

preparing output

The `showInputDialog` method in lines 6–7 displays an input dialog. Enter the interest rate as a double value and click *OK* to accept the input. The value is returned as a string that is assigned to the `String` variable `annualInterestRateString`. The `Double.parseDouble(annualInterestRateString)` (line 11) is used to convert the string into a `double` value. If you entered an input other than a numeric value or clicked *Cancel* in the input dialog box, a runtime error would occur. In Chapter 13, “Exception Handling,” you will learn how to handle the exception so that the program can continue to run.



Pedagogical Note

For obtaining input you can use `JOptionPane` or `Scanner`, whichever is convenient. For consistency most examples in this book use `Scanner` for getting input. You can easily revise the examples using `JOptionPane` for getting input.

`JOptionPane` or `Scanner`?

KEY TERMS

algorithm	24	data type	25
assignment operator (=)	30	debugger	55
assignment statement	30	debugging	55
backslash (\)	46	declaration	30
<code>byte</code> type	27	decrement operator (--)	41
casting	41	<code>double</code> type	33
<code>char</code> type	44	encoding	45
constant	31	<code>final</code>	31