

Using JOptionPane

For Introduction to Java Programming
By Y. Daniel Liang

1 Introduction

`JOptionPane` contains the static methods for displaying dialog boxes for input and output. It is a Swing component, but you can call these methods with any Java program including JavaFX programs. This supplement gives a brief introduction on using `JOptionPane` for simple output and input.

2 Displaying Text in a Message Dialog Box

(NOTE: This section may be covered after Chapter 1.)

The program in Listing 1.1 in the textbook displays the text on the console. You can rewrite the program to display the text in a message dialog box. To do so, you need to use the `showMessageDialog` method in the `JOptionPane` class. `JOptionPane` is one of the many predefined classes in the Java library that you can reuse rather than “reinvent the wheel.” You can use the `showMessageDialog` method to display any text in a message dialog box, as shown in [Figure 1](#). The new program is given in [Listing 1](#).

<margin note> `JOptionPane`

<margin note> `showMessageDialog`

Figure 1

“Welcome to Java!” is displayed in a message box.



Listing 1 WelcomeInMessageDialogBox.java

```
1  /* This application program displays Welcome to Java!  
2   * in a message dialog box.  
3   */  
4  import javax.swing.JOptionPane;  
5  
6  public class WelcomeInMessageDialogBox {  
7      public static void main(String[] args) {  
8          // Display Welcome to Java! in a message dialog box  
9          JOptionPane.showMessageDialog(null, "Welcome to Java!");  
10     }  
11 }
```

*<margin note (line 1)>*block comment

*<margin note (line 4)>*import

*<margin note (line 7)>*main method

*<margin note (line 9)>*display message

<end Listing 1>

The first three lines are block comments. The first line begins with `/*` and the last line ends with `*/`. By convention, all other lines begin with an asterisk (`*`).

This program uses the Java class `JOptionPane` (line 9). Java's predefined classes are grouped into **packages**. `JOptionPane` is in the `javax.swing` package. `JOptionPane` is imported into the program using the `import` statement in line 4 so that the compiler can locate the class without the full name

`javax.swing.JOptionPane`.

*<margin note>*package

<box>

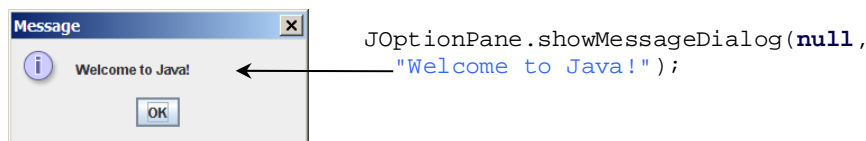
NOTE

If you replace `JOptionPane` in line 9 with `javax.swing.JOptionPane`, you don't need to import it in line 4. `javax.swing.JOptionPane` is the full name for the `JOptionPane` class.

<end box>

The `showMessageDialog` method is a *static* method. Such a method should be invoked by using the class name followed by a dot operator (`.`) and the method name with arguments. Details of methods will be discussed in **Chapter 5**. The

`showMessageDialog` method can be invoked with two arguments, as shown below.



The first argument can always be `null`. `null` is a Java keyword that will be fully discussed in **Chapter 9**. The second argument is a string for text to be displayed.

There are several ways to use the `showMessageDialog` method. For the time being, you need to know only two ways. One is to use a statement, as shown in the example:


*<margin note>*two versions of `showMessageDialog`

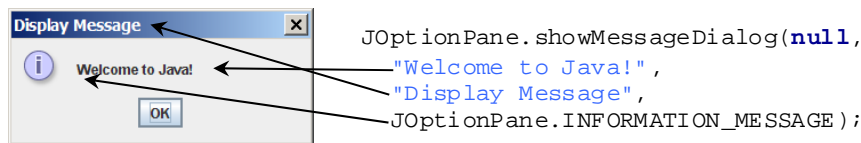
```
JOptionPane.showMessageDialog(null, x);
```

where `x` is a string for the text to be displayed.

The other is to use a statement like this one:

```
JOptionPane.showMessageDialog(null, x,  
y, JOptionPane.INFORMATION_MESSAGE);
```

where `x` is a string for the text to be displayed, and `y` is a string for the title of the message box. The fourth argument can be `JOptionPane.INFORMATION_MESSAGE`, which causes the information icon () to be displayed in the message box, as shown in the following example.



<box>

NOTE

There are two types of `import` statements: *specific import* and *wildcard import*. The *specific import* specifies a single class in the import statement. For example, the following statement imports `JOptionPane` from the package `javax.swing`.

*<key term>*specific import

```
import javax.swing.JOptionPane;
```

The **wildcard import** imports all the classes in a package by using the asterisk as the wildcard. For example, the following statement imports all the classes from the package `javax.swing`.

*<key term>*wildcard import

```
import javax.swing.*;
```

The information for the classes in an imported package is not read in at compile time or runtime unless the class is used in the program. The import statement simply tells the compiler where to locate the classes. There is **no performance difference** between a specific import and a wildcard import declaration.

*<margin note>*no performance difference

<end box>

<box>

NOTE

Recall that you have used the `System` class in the statement

```
System.out.println("Welcome to Java"); in Listing 1.1. The
```

`System` class is not imported because it is in the **java.lang** package. All the

classes in the **java.lang** package are **implicitly imported** in every Java

program.

*<margin note>*java.lang package

*<margin note>*implicitly imported

<end box>

3 Getting Input from Input Dialogs

(NOTE: This section may be covered after Chapter 2.)

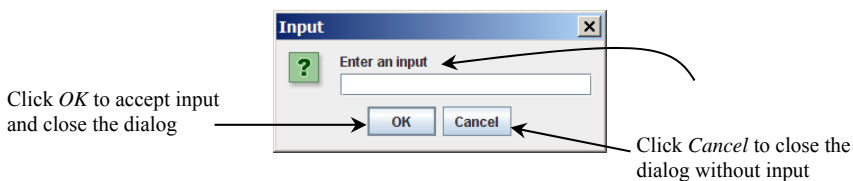
You can obtain input from the console. Alternatively, you can obtain input from an input dialog box by invoking the `JOptionPane.showInputDialog` method, as shown in

Figure 2.

<margin note> *JOptionPane class*

Figure 2

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



```
String input =  
    JOptionPane.showInputDialog(  
        "Enter an input");
```

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 close 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.

<margin note> *showInputDialog method*

One is to use a statement like this one:

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

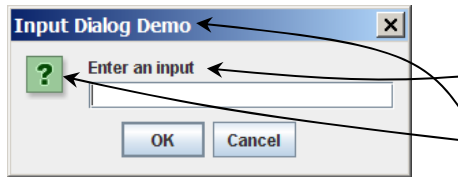
where **x** is a string for the prompting message.

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.



```
String input =  
JOptionPane.showInputDialog(null,  
"Enter an input",  
"Input Dialog Demo",  
JOptionPane.QUESTION_MESSAGE);
```

3.1 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.

To convert a string into an **int** value, use the **Integer.parseInt** method, as follows:

<margin note> *Integer.parseInt* method

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

where **intString** is a numeric string such as **123**.

To convert a string into a **double** value, use the **Double.parseDouble** method, as follows:

<margin note> *Double.parseDouble* method

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

where **doubleString** is a numeric string such as **123.45**.

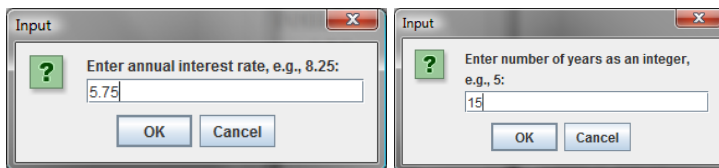
The `Integer` and `Double` classes are both included in the `java.lang` package, and thus they are automatically imported.

3.2 Using Input Dialog Boxes

Having learned how to read input from an input dialog box, you can rewrite the program in Listing 2.9, `ComputeLoan.java` in the textbook, to read from input dialog boxes rather from the console. Listing 2 gives the complete program. Figure 3 shows a sample run of the program.

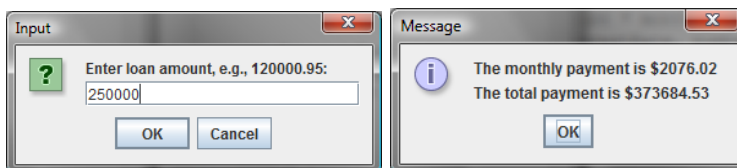
Figure 3

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).



(a)

(b)



(c)

(d)

Listing 2 `ComputeLoanUsingInputDialog.java`

```
1 import javax.swing.JOptionPane;  
2
```



```

3 public class ComputeLoanUsingInputDialog {
4     public static void main(String[] args) {
5         // Enter annual interest rate
6         String annualInterestRateString = JOptionPane.showInputDialog(
7             "Enter annual interest rate, for example, 8.25:");
8
9         // Convert string to double
10        double annualInterestRate =
11            Double.parseDouble(annualInterestRateString);
12
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, for 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            "\n\nThe total payment is $" + totalPayment;
42        JOptionPane.showMessageDialog(null, output);
43    }
44 }

```

<margin note on (line 6)>enter interest rate

<margin note on (line 10)>convert string to double

<margin note on (line 31)>monthlyPayment

<margin note on (line 33)>totalPayment

<margin note on (line 36)>preparing output

<end Listing 2>

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.

<box>

Pedagogical NOTE

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

<margin note>*JOptionPane or Scanner?*

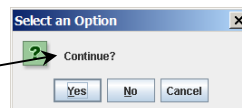
<end box>

4 Confirmation Dialogs

(NOTE: This section may be covered after Chapter 4.)

You have used `showMessageDialog` to display a message dialog box and `showInputDialog` to display an input dialog box. Occasionally it is useful to answer a question with a confirmation dialog box. A confirmation dialog can be created using the following statement:

```
int option =  
    JOptionPane.showConfirmDialog  
        (null, "Continue?");
```



When a button is clicked, the method returns an option value. The value is `JOptionPane.YES_OPTION` (0) for the *Yes* button, `JOptionPane.NO_OPTION` (1) for the *No* button, and `JOptionPane.CANCEL_OPTION` (2) for the *Cancel* button.

You may rewrite the guess-birthday program in Listing 4.3 in the textbook using confirmation dialog boxes, as shown in Listing 3. Figure 3 shows a sample run of the program for the day 19.

Listing 3 GuessBirthdayUsingConfirmationDialog.java

```
1  import javax.swing.JOptionPane;
2
3  public class GuessBirthdayUsingConfirmationDialog {
4      public static void main(String[] args) {
5          String set1 =
6              " 1  3  5  7\n" +
7              " 9 11 13 15\n" +
8              "17 19 21 23\n" +
9              "25 27 29 31";
10
11         String set2 =
12             " 2  3  6  7\n" +
13             "10 11 14 15\n" +
14             "18 19 22 23\n" +
15             "26 27 30 31";
16
17         String set3 =
18             " 4  5  6  7\n" +
19             "12 13 14 15\n" +
20             "20 21 22 23\n" +
```

```
21     "28 29 30 31";
22
23     String set4 =
24         " 8  9 10 11\n" +
25         "12 13 14 15\n" +
26         "24 25 26 27\n" +
27         "28 29 30 31";
28
29     String set5 =
30         "16 17 18 19\n" +
31         "20 21 22 23\n" +
32         "24 25 26 27\n" +
33         "28 29 30 31";
34
35     int day = 0;
36
37     // Prompt the user to answer questions
38     int answer = JOptionPane.showConfirmDialog(null,
39         "Is your birthday in these numbers?\n" + set1);
40
41     if (answer == JOptionPane.YES_OPTION)
42         day += 1;
43
44     answer = JOptionPane.showConfirmDialog(null,
45         "Is your birthday in these numbers?\n" + set2);
46
47     if (answer == JOptionPane.YES_OPTION)
48         day += 2;
49
50     answer = JOptionPane.showConfirmDialog(null,
51         "Is your birthday in these numbers?\n" + set3);
52
```

```

53     if (answer == JOptionPane.YES_OPTION)
54         day += 4;
55
56     answer = JOptionPane.showConfirmDialog(null,
57         "Is your birthday in these numbers?\n" + set4);
58
59     if (answer == JOptionPane.YES_OPTION)
60         day += 8;
61
62     answer = JOptionPane.showConfirmDialog(null,
63         "Is your birthday in these numbers?\n" + set5);
64
65     if (answer == JOptionPane.YES_OPTION)
66         day += 16;
67
68     JOptionPane.showMessageDialog(null, "Your birthday is " +
69         day + "!");
70 }
71 }

```

*<margin note (line 1)>*import class

*<margin note (line 5)>*set1

*<margin note (line 11)>*set2

*<margin note (line 17)>*set3

*<margin note (line 23)>*set4

*<margin note (line 29)>*set5

*<margin note (line 38)>*confirmation dialog

*<margin note (line 41)>*in set1?

*<margin note (line 47)>*in set2?

*<margin note (line 53)>*in set3?

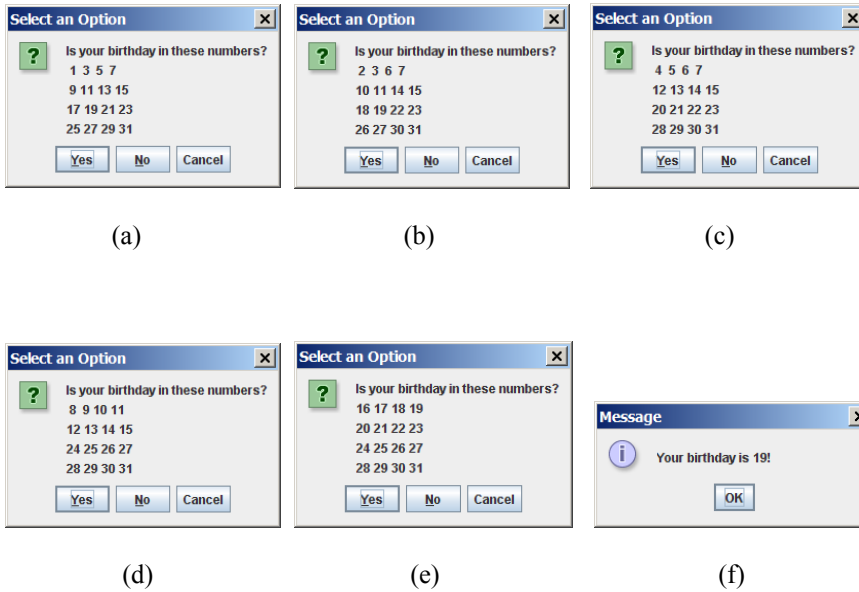
*<margin note (line 59)>*in set4?

*<margin note (line 65)>*in set5?

<end Listing 3>

Figure 3

Click Yes in (a), Yes in (b), No in (c), No in (d), and Yes in (e).



The program displays confirmation dialog boxes to prompt the user to answer whether a number is in Set1 (line 38), Set2 (line 44), Set3 (line 50), Set4 (line 56), and Set5 (line 62). If the answer is Yes, the first number in the set is added to `day` (lines 42, 48, 54, 60, and 66).