19. GUI Basics

Java

Summer 2008

Instructor: Dr. Masoud Yaghini

Outline

- Displaying Text in a Message Dialog Box
- Getting Input from Input Dialogs
- References

Displaying Text in a Message Dialog Box

Displaying Text in a Message Dialog Box

- You can use the showMessageDialog method in the JOptionPane class to display any text in a message dialog box.
- JOptionPane class is in the javax.swing package.

WelcomeInMessageDialogBox.java

```
1  /** This application program displays Welcome to Java!
2   *  in a message dialog box.
3   */
4  package chapter01; // Organize the source files into packages
5
6  import javax.swing.JOptionPane;
7
8  public class WelcomeInMessageDialogBox {
9    public static void main(String[] args) {
10     // Display Welcome to Java! in a message dialog box
11     JOptionPane.showMessageDialog(null, "Welcome to Java!",
12         "Display Message", JOptionPane.INFORMATION_MESSAGE);
13   }
14 }
```

WelcomeInMessageDialogBox.java

- "Welcome to Java!" is displayed in a message box.

WelcomeInMessageDialogBox.java

- JOptionPane is imported to the program using the import statement so that the compiler can locate the class.
- If you replace JOptionPane with javax.swing.JOptionPane, you don't need to import it

WelcomeInMessageDialogBox.java

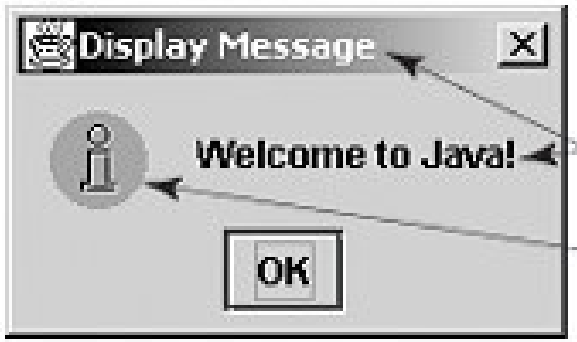
- The System class has used in the statement System.out.println("Welcome to Java");
- The System class is not imported because it is in the java.lang package.
- All the classes in the java.lang package are implicitly imported in every Java program.

showMessageDialog method

- 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.
- showMessageDialog method can be invoked with four arguments

showMessageDialog method

- The first argument can always be null. null is a Java keyword that will be fully introduced later.
- The second argument can be a string for text to be displayed.
- The third argument is the title of the message box.
- The fourth argument can be JOptionPane.INFORMATION_MESSAGE, which causes the icon to be displayed in the message box.




```
JOptionPane.showMessageDialog(null,
  "Welcome to Java!",
  "Display Message",
  JOptionPane.INFORMATION_MESSAGE);
```

showMessageDialog method

- The other way to use showMessageDialog method is:

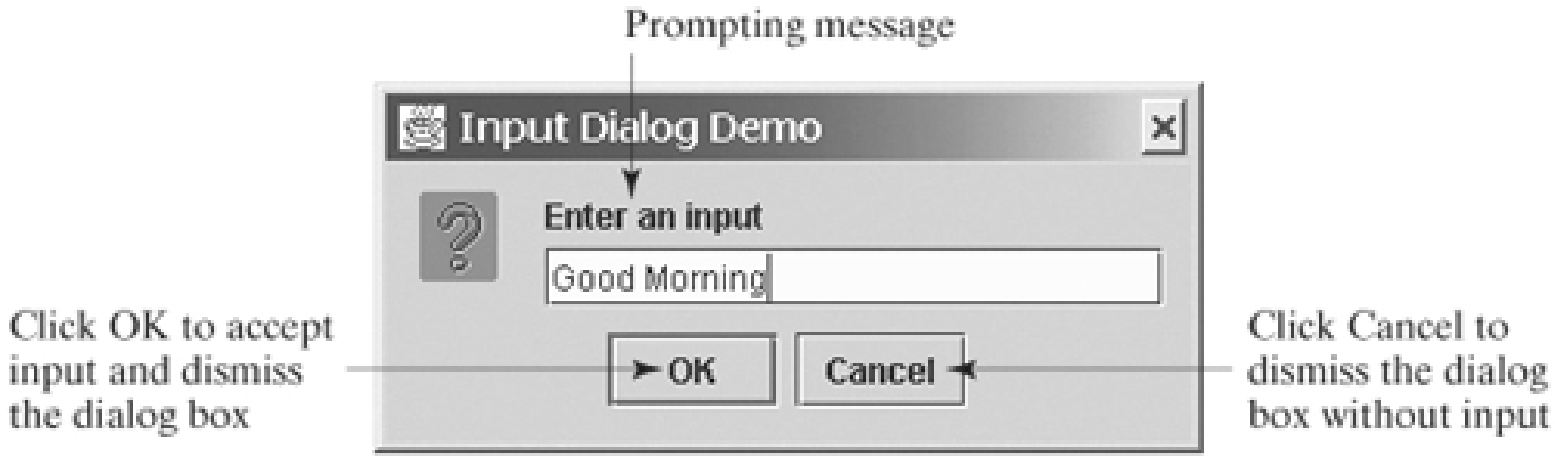
```
JOptionPane.showMessageDialog(null, "Welcome to Java!");
```

- The output:

Getting Input from Input Dialogs Box

Getting Input from Input Dialogs

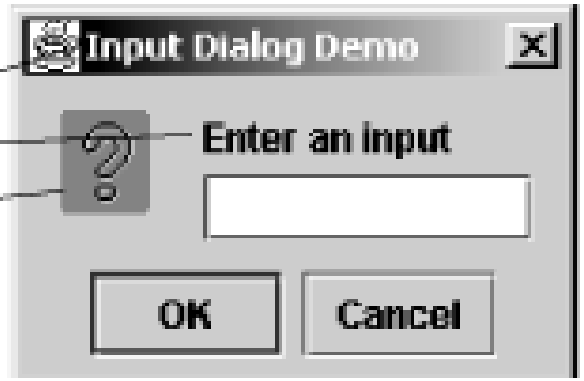
- You can use the showInputDialog method in the JOptionPane class to get input at runtime.
- When this method is executed, a dialog is displayed to enable you to enter a string.



Getting Input from Input Dialogs

- 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.
- You can invoke the method with four arguments, as follows:

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



Getting Input from Input Dialogs

- The first argument can always be null.
- The second argument is a string that prompts the user.
- The third argument is the title of the input box.
- The fourth argument can be JOptionPane.QUESTION_MESSAGE, which causes the question icon to be displayed in the input box.

Getting Input from Input Dialogs

- The other way to use a statement like this one:

  JOptionPane.showInputDialog(x);

- Where x is a string for the prompting message.

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 parseInt method in the Integer class, as follows:

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

- Where intString is a numeric string such as "123".

Converting Strings to Numbers

- 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 are automatically imported.

ComputeLoan.java

- This example shows you how to write a program that computes loan payments.
- The program lets the user enter the interest rate, number of years, and loan amount, and then computes the monthly payment and the total payment.
- It concludes by displaying the monthly and total payments.
- The formula to compute the monthly payment is as follows:

$$\frac{\text{loanAmount} \times \text{monthlyInterestRate}}{1 - \frac{1}{(1 + \text{monthlyInterestRate})^{\text{numberOfYears} \times 12}}}$$

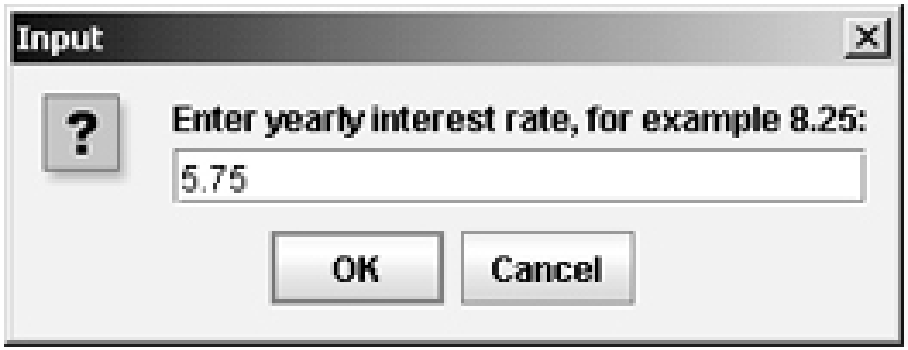
ComputeLoan.java

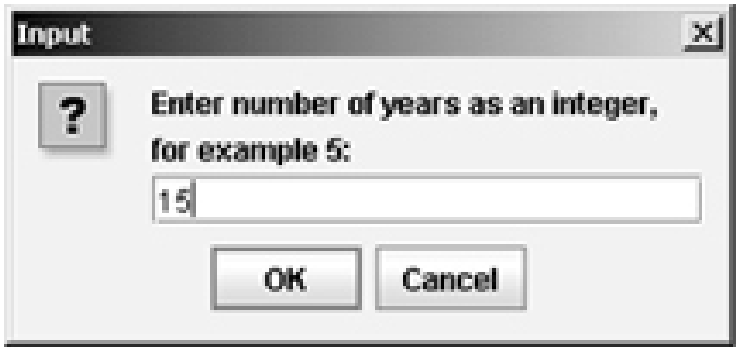
```
1  package chapter02;
2
3  import javax.swing.JOptionPane;
4
5  public class ComputeLoan {
6    /** Main method */
7    public static void main(String[] args) {
8      // Enter yearly interest rate
9      String annualInterestRateString = JOptionPane.showInputDialog(
10         "Enter yearly interest rate, for example 8.25:");
11
12     // Convert string to double
13     double annualInterestRate = Double.parseDouble(annualInterestRateString);
14
15     // Obtain monthly interest rate
16     double monthlyInterestRate = annualInterestRate / 1200;
17
18     // Enter number of years
19     String numberOfYearsString = JOptionPane.showInputDialog(
20         "Enter number of years as an integer, \nfor example 5:");
21
22     // Convert string to int
23     int numberOfYears = Integer.parseInt(numberOfYearsString);
```

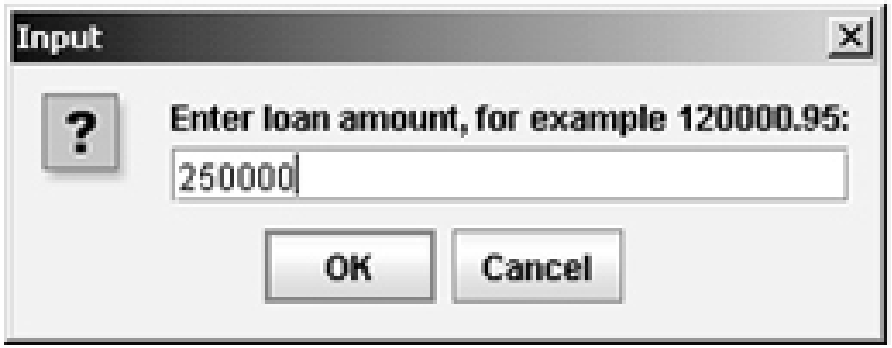
ComputeLoan.java

```
24
25        // Enter loan amount
26        String loanString = JOptionPane.showInputDialog(
27              "Enter loan amount, for example 120000.95:");
28
29        // Convert string to double
30        double loanAmount = Double.parseDouble(loanString);
31
32        // Calculate payment
33        double monthlyPayment = loanAmount * monthlyInterestRate / (1
34              - 1 / Math.pow(1 + monthlyInterestRate, numberOfYears * 12));
35        double totalPayment = monthlyPayment * numberOfYears * 12;
36
37        // Format to keep two digits after the decimal point
38        monthlyPayment = (int)(monthlyPayment * 100) / 100.0;
39        totalPayment = (int)(totalPayment * 100) / 100.0;
40
41        // Display results
42        String output = "The monthly payment is " + monthlyPayment +
43              "\nThe total payment is " + totalPayment;
44        JOptionPane.showMessageDialog(null, output);
45    }
46 }
```

ComputeLoan.java







- If you click Cancel or enter some letters instead of numbers in the input dialog box, a runtime error would occur.

References

References

- Y. Daniel Liang, **Introduction to Java Programming**, Sixth Edition, Pearson Education, 2007. (Chapter 1 & 2)

The End