19. GUI Basics

Java

Fall 2009

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

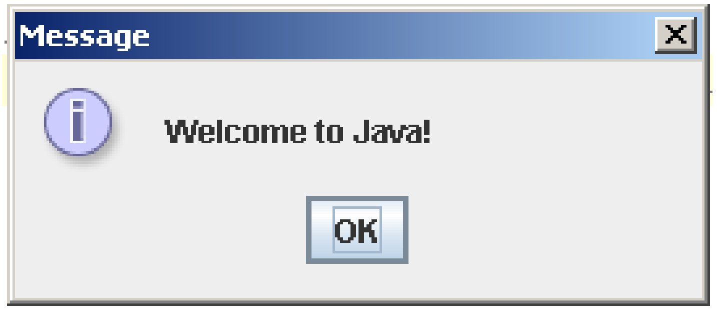
- You can use the showMessageDialog method in the JOptionPane class to display any text in a message dialog box.
- JOptionPane class is in the javax.swing package.
- Example:
  - WelcomeInMessageDialogBox.java

showMessageDialog method

- The other way to use showMessageDialog method is:

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

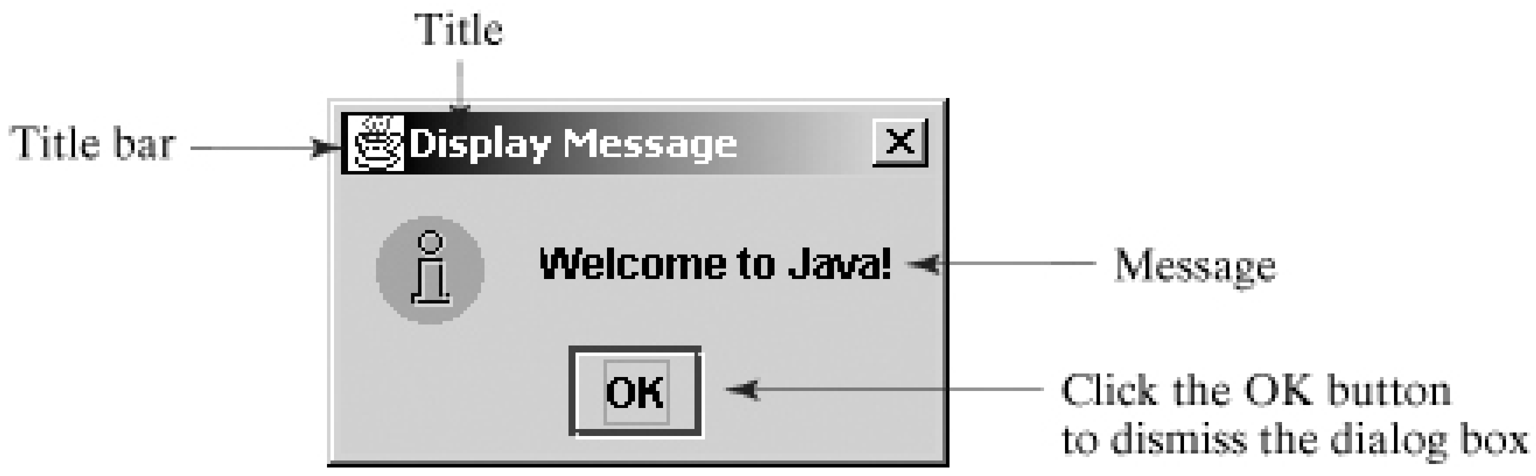
- The output:



Welcome to Java!

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

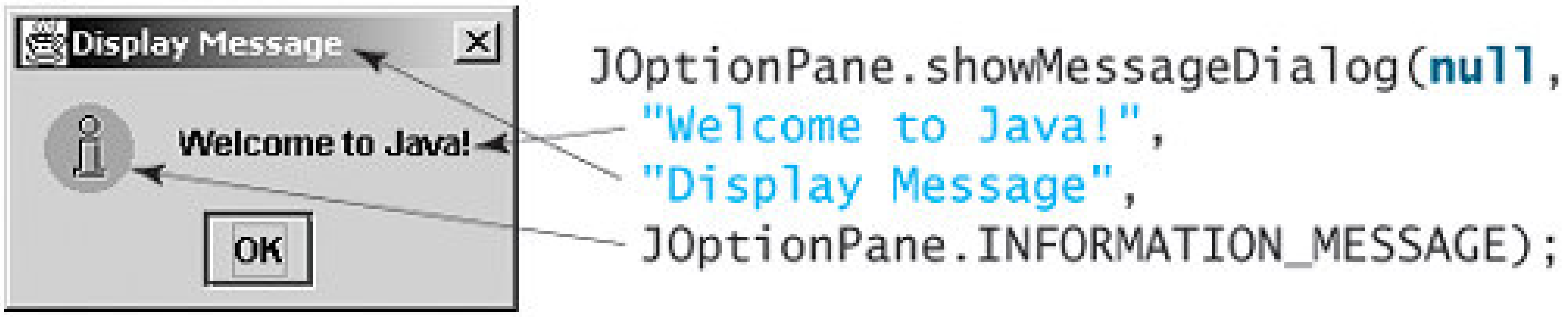
- The System class has used in the statement System.out.println("Welcome to Java");
- The System class is not imported because it is in the java.lang package.
- All the classes in the java.lang package are implicitly imported in every Java program.

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.

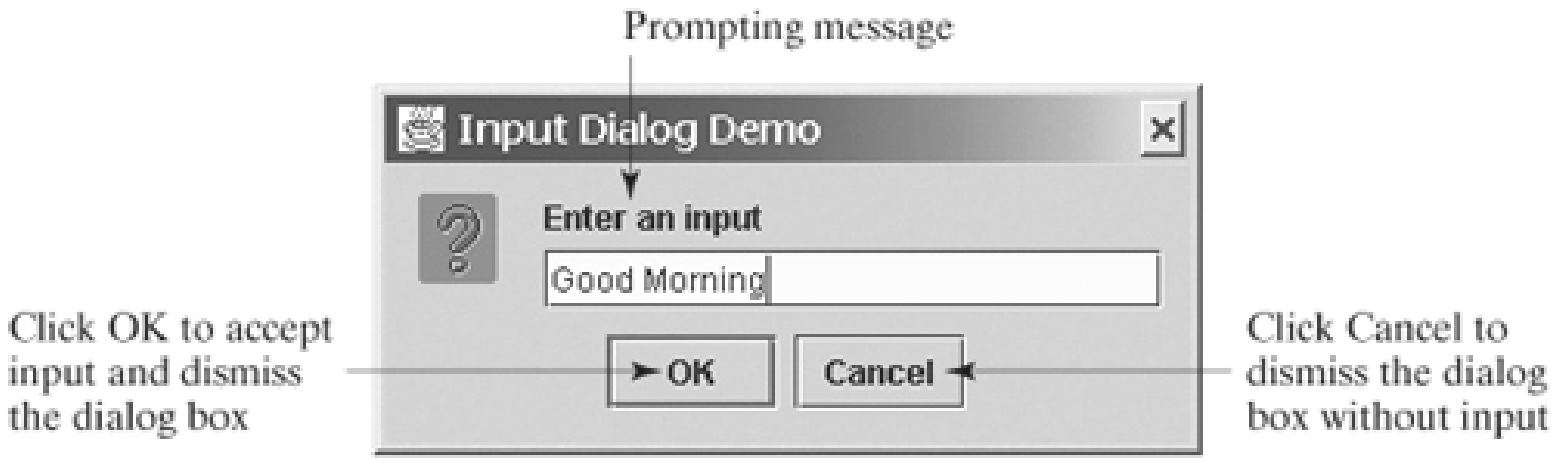


```
JOptionPane.showMessageDialog(null,
  "Welcome to Java!",
  "Display Message",
  JOptionPane.INFORMATION_MESSAGE);
```

Getting Input from Input Dialogs Box

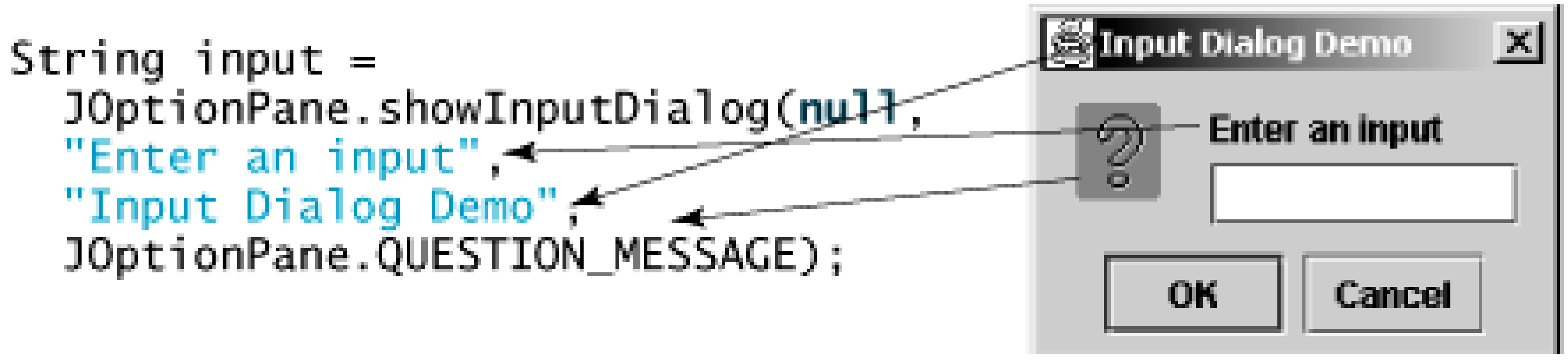
Getting Input from Input Dialogs

- 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.


Prompting message
Input Dialog Demo
Enter an input
Good Morning
OK
Cancel
Click OK to accept input and dismiss the dialog box
Click Cancel to dismiss the dialog box without input

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.

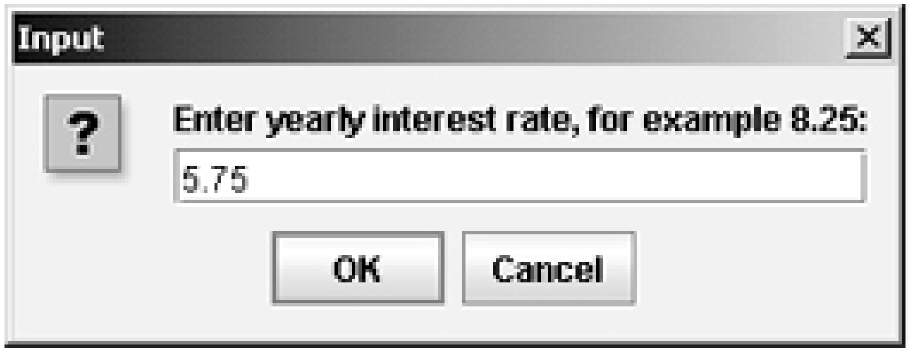
ComputeLoan.java

- The formula to compute the monthly payment is as follows:

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

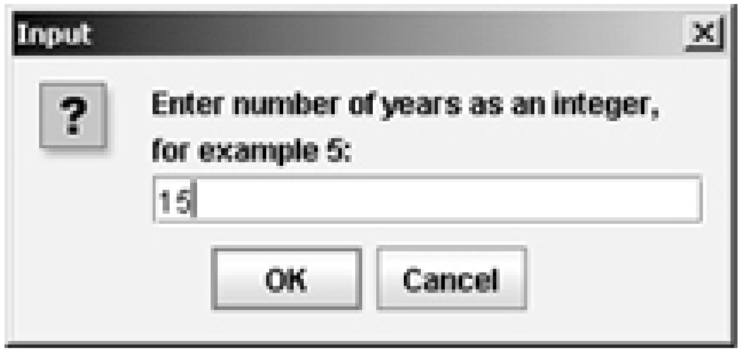
- Program:
  - ComputeLoan.java

ComputeLoan.java



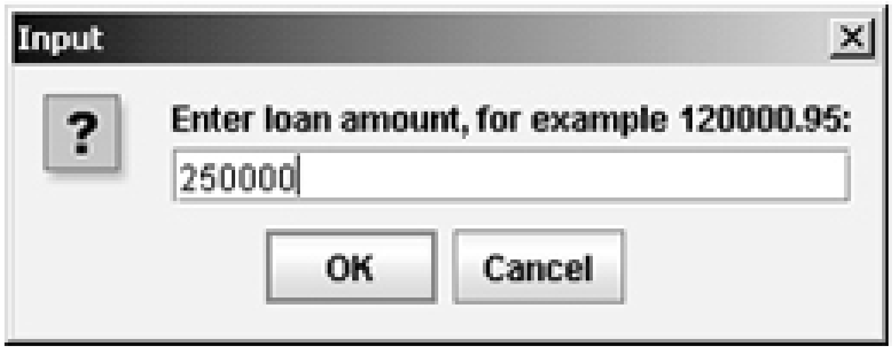
Enter yearly interest rate, for example 8.25:

5.75



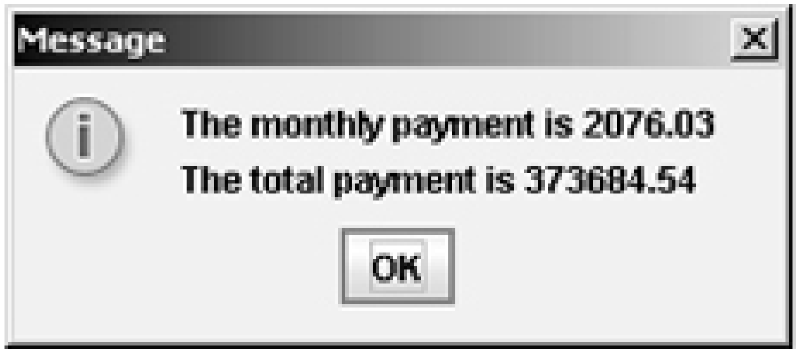
Enter number of years as an integer, for example 5:

15



Enter loan amount, for example 120000.95:

250000



The monthly payment is 2076.03
The total payment is 373684.54

- 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