GUI methods for I/O

JOptionPane: an input/output class

- Most Java programs employ a graphical user interface (GUI) with two types of windows:
  - frame window: the type of window that displays most programs
  - dialog window: allows communication with user; most commonly used to prompt for & receive input, but we’ll begin by looking at just output

- Example:
  ```java
  JOptionPane.showMessageDialog(null, "Look at me!");
  ```
  - displays a small window in the center of the screen containing the words in the string literal and a button labeled “OK”
  - this is an example of a message that calls a class method of the JOptionPane class – we are not making a request to an object, but rather to the class itself
    - recall the syntax we saw before for a message:
      ```java
      objectName.methodName(argument(s));
      ```
    - this new syntax is a variation:
      ```java
      className.methodName(argument(s));
      ```
    - when we start defining our own classes and methods, you will see how these two mechanisms work
  - arguments to the method:
    - the first argument, null, indicates that there is no existing frame object upon which this dialog should appear; if we want the dialog to appear in an existing frame window, we would pass the name of the frame object as the first argument, instead of null – for example:
    - the second argument, a string literal, indicates the text we want to display in the window

- Displaying multiple lines of text: the special character ‘\n’ represents the control character you get when you press the Enter key on the keyboard (n = new line)

- Later on, we’ll introduce another JOptionPane class method that allows us to take keyboard input from the user

Reading Input

In order to receive input data from a user, we need two things:

- a mechanism by which to read the data
- a place to store the data

We can use another method, showInputDialog, of the JOptionPane class for the first part; we can use a String object for the second part.

The showInputDialog method:
The syntax for showInputDialog is almost identical to the previous JOptionPane method we looked at, showMessageDialog. You may recall from a previous example program the following lines of code:

```java
JOptionPane.showMessageDialog(myWindow, "It’s my window");
JOptionPane.showMessageDialog(null, "and I’ll cry \n if I want to");
```

In general, the syntax for both showMessageDialog and showInputDialog is:
JOptionPane.methodName (WindowObject, MessageObject);

We know from the examples that the first argument, the WindowObject, can either be a JFrame object we have declared and initialized or null; we also know that the second argument can be a String literal value.

The difference between the two methods is that showMessageDialog merely displays a window containing the specified message, but showInputDialog method displays the message as a prompt, followed by a space for the user to enter input. For example, the following code fragment produces a dialog window like the one shown below:

```java
String daddy;
daddy = JOptionPane.showInputDialog (null, “Who’s your daddy?”);
```

Reading Numeric Input

- We have already seen that we can read Strings via a showInputDialog window
- It turns out we can use the same mechanism to read other kinds of data, but it takes a little more work
- In order to transform the String returned by showInputDialog into a number, we must perform a type conversion
- Type conversion of a string to a numeric type is accomplished using various parse methods, which are members of the numeric wrapper classes
- Wrapper classes exist as a bridge between simple types (like int or double) and object types (like String)
- The following code fragment illustrates the mechanism:
  ```java
  String input;
  int userAge;
  input = JOptionPane.showInputDialog (null, “Enter your age in years”);
  userAge = Integer.parseInt(input);
  ```
- In the example above, the wrapper class is Integer, and the parse method is parseInt
- Some other useful wrapper classes include Long, Float and Double, and their corresponding parse methods, parseLong, parseFloat and parseDouble
- The general syntax for conversion of a String to a number is:
  ```java
  WrapperClassName.parseMethod(String)
  ```
  where WrapperClassName is Integer, Double, etc. and parseMethod is parseInt or parseDouble, etc. and String is a String object containing the text to be converted