How GUI Programs Talk to an Application Object

Introduction

Programs that use a Graphical User Interface typically contain one or more classes that represent application objects and that do the application work. For this example, consider the Box class that was show to you in lab #12. Recall that this class represents a box with a height, a width, and a depth.

Defining the Roles

When you create a program of this kind, it is important to understand the role of each class involved. The role of the Form class is to manage the user interace. In particular, the Form class provides user interface components like buttons, text boxes and menu items, and the methods that we write to react to events that are generated when users interact with these components. On the other hand, the role of the Box class is to define the data that defines a Box, and provide the methods to manage this data. In this case, the Box class only needs to keep track of three pieces of data: the height of a box, the width of a box, and the depth of a box. We need a few methods to manage this data:

1. A default constructor that sets each variable in the object to zero.
2. A parameterized constructor that sets these variables to values passed in as parameters to the method.
3. A method to tell us what the height of the box is.
4. A method to tell us what the width of the box is.
5. A method to tell us what the depth of the box is.
6. A method to tell us what the volume of the box is.

As you study the code for this program, note how the event handling code in the Form class has to call on the Box object to do the real work of the application. When one object calls a method in another object, we say that it sends the other object a message. Also notice how the code in the Box class is completely independent of the user interface. It knows nothing about users or buttons or text boxes.

The code

The source code for the Form class is here.

The source code for the Box class is here.

You can get an executable for this program here.