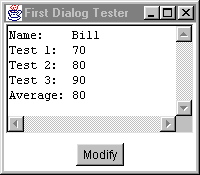
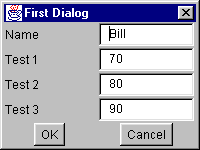
**Dialogs**

A ***dialog*** is a window that pops up during the run of an application. Examples are message boxes and file dialogs. Message boxes merely display information, whereas file dialogs take information from the user and return it to the application. Normally, a dialog is ***modal*** in that the rest of the application is inaccessible as long as the dialog is active. The user quits most dialogs by selecting an **OK** button or a **Cancel** button. The **Cancel** button allows users to back out of changes that are not really desired.

Normally, Java programmers must extend the Dialog class in order to construct dialogs. However, extending BreezyGUI's GBDialog class provides a much easier alternative.

The next program shows how to construct a dialog by extending GBDialog. The application, FirstDialogTester, displays the information for a student. There is a single command button called **Modify**. When the user selects this button, a dialog window pops up, as shown below:

Note that the application window on the left has become inactive. The dialog on the right allows the user to change the student's grades and to return to the application by selecting either **OK** or **Cancel**. If the user selects **Cancel**, the information for the student in the application remains unchanged; if the user selects **OK**, the information in the application is updated with the information from the dialog. When the user selects either one of these buttons, the dialog disappears and the application window becomes active again.

The demo consists of three classes:

|  | Student − a simple class that defines a student with a name and three test scores |
| --- | --- |
|  | FirstDialog − the module for the dialog |
|  | FirstDialogTester − the application that tests the dialog. |

We first present the code for FirstDialogTester and then discuss what it does:

import java.awt.\*;

import BreezyGUI.\*;

public class FirstDialogTester extends GBFrame{

TextArea taStudent = addTextArea ("",1,1,2,3);

Button btModify = addButton ("Modify",4,1,2,1);

private Student stud;

public FirstDialogTester(){

stud = new Student ("Bill", 70, 80, 90);

taStudent.setText (stud.toString());

setTitle ("First Dialog Tester");

}

public void buttonClicked (Button buttonObj){

FirstDialog dlg;

dlg = new FirstDialog (this, stud);

dlg.show();

if (dlg.getDlgCloseIndicator().equals("OK"))

taStudent.setText (stud.toString());

}

public static void main (String[] args){

Frame frm = new FirstDialogTester();

frm.setSize (200, 175);

frm.setVisible (true);

}

}

When this program starts up, it creates a new Student object with some information and displays this information in a text area. The program then waits for the user to click the **Modify** button. When the user does this, buttonClicked

|  | creates a new instance of FirstDialog, passing to it the parameters this (the application itself) and the student |
| --- | --- |
|  | sends the show message to the dialog; this will pop up the dialog and pass control to it |
|  | After the dialog is closed, execution resumes at the statement following show, at which point the application determines the manner in which the user closed the dialog and then takes the appropriate action. |

Here is the code for the FirstDialog class (the embedded comments make the code fairly self-explanatory):

import java.awt.\*;

import BreezyGUI.\*;

public class FirstDialog extends GBDialog{

// Layout the GUI in the usual manner

private Label lbName = addLabel ("Name" ,1,1,1,1);

private Label lbTest1 = addLabel ("Test 1",2,1,1,1);

private Label lbTest2 = addLabel ("Test 2",3,1,1,1);

private Label lbTest3 = addLabel ("Test 3",4,1,1,1);

private TextField tfName = addTextField ("" ,1,2,1,1);

private IntegerField ifTest1 = addIntegerField (0 ,2,2,1,1);

private IntegerField ifTest2 = addIntegerField (0 ,3,2,1,1);

private IntegerField ifTest3 = addIntegerField (0 ,4,2,1,1);

private Button btnOK = addButton ("OK" ,5,1,1,1);

private Button btnCancel = addButton ("Cancel",5,2,1,1);

// The dialog is passed a student object and must refer to it

// at several locations thereafter.

private Student stud;

public FirstDialog (Frame f, Student s){

// The next few lines are part of every dialog

super (f); // \*\*\* REQUIRED

setTitle ("First Dialog"); // Optional, but

// the default is ""

setDlgCloseIndicator ("Cancel"); // Optional, because

// the default is "Cancel"

setSize (200, 150); // Optional, but

// the default is 300,300

// The remaining lines are application dependent

stud = s;

tfName.setText (stud.getName());

ifTest1.setNumber (stud.getScore(1));

ifTest2.setNumber (stud.getScore(2));

ifTest3.setNumber (stud.getScore(3));

}

public void buttonClicked (Button buttonObj){

// The choice of buttons is entirely up to the programmer

// HOWEVER, pull down menus are NOT available.

if (buttonObj == btnOK){

stud.setName (tfName.getText());

stud.setScore (1, ifTest1.getNumber());

stud.setScore (2, ifTest2.getNumber());

stud.setScore (3, ifTest3.getNumber());

setDlgCloseIndicator ("OK");

// \*\*\* Very important line otherwise the application

// will think the user cancelled.

}

dispose(); // This is the way one closes a dialog

}

}

Here is the code for the simple student class:

public class Student extends Object{

// Instance variables

// Each student object will have a name and three test scores

private String name;

private int test1;

private int test2;

private int test3;

// Constructor method

// Initialize a new student's name to the empty string and his test

// scores to zero.

public Student(){

name = "";

test1 = 0;

test2 = 0;

test3 = 0;

}

// Other methods

// Set a student's name

public void setName (String nm){

name = nm;

}

// Get a student's name

public String getName (){

return name;

}

// Set the score on the indicated test

public void setScore (int i, int score){

if (i == 1) test1 = score;

else if (i == 2) test2 = score;

else test3 = score;

}

// Get the score on the indicated test

public int getScore (int i){

if (i == 1) return test1;

else if (i == 2) return test2;

else return test3;

}

// Compute and return a student's average

public int getAverage(){

int average;

average = (int) Math.round((test1 + test2 + test3) / 3.0);

return average;

}

// Compute and return a student's highest score

public int getHighScore(){

int highScore;

highScore = test1;

if (test2 > highScore) highScore = test2;

if (test3 > highScore) highScore = test3;

return highScore;

}

// Return a string representation of a student's name, test scores

// and average.

public String toString(){

String str;

str = "Name: " + name + "\n" + // "\n" denotes a newline

"Test 1: " + test1 + "\n" +

"Test 2: " + test2 + "\n" +

"Test 3: " + test3 + "\n" +

"Average: " + getAverage();

return str;

}

}