**Menus and Window Titles**

***Menus***

***Menus*** provide a convenient mechanism for entering commands into a program. A menu system consists of a menu bar, a number of menus, and, for each menu, several selections. It is also possible to have submenus, but we ignore these for now. It is easy to add a menu system to an application. Simply declare a menu item object for each menu selection. For instance, the following code adds two menus to an application’s interface:

MenuItem highTest1MI = addMenuItem ("HighStudent", "Test1");

MenuItem highTest2MI = addMenuItem ("HighStudent", "Test2");

MenuItem highTest3MI = addMenuItem ("HighStudent", "Test3");

MenuItem highOverallMI = addMenuItem ("HighStudent", "Overall");

MenuItem highAverageMI = addMenuItem ("HighStudent", "Average");

MenuItem displayStudent1MI = addMenuItem ("Display", "Student1");

MenuItem displayStudent2MI = addMenuItem ("Display", "Student2");

The first menu is called **HighStudent** and has five items, while the second, called **Display**, has two. When the user selects a menu item, a method called menuItemSelected is activated. This method can be placed either before or after the buttonClicked method. Here is a typical snippet of code from a menuItemSelected method:

public void menuItemSelected (MenuItem menuItemObj){

if (menuItemObj == highTest1MI)

... do something appropriate ...

else if (menuItemObj == highTest2MI){

... do something else ...

etc.

}

***The setTitle Method***

Most applications include a title at the top of the window. To display a title in our applications, we need to add a constructor to the interface class and include the line:

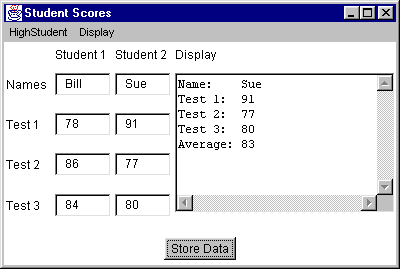
setTitle (<"the title">);

***Example: Using Menus and a Window Title***

Here is an example that illustrates the use of menus and window titles.

The Interface

The interface for the example program looks like this:



The menus consist of:

HighStudent

Test1 Display the name of the student who scored higher on test 1

Test2 Display the name of the student who scored higher on test 2

Test3 Display the name of the student who scored higher on test 3

Overall Display the name of the student who had the highest test score

Average Display the name of the student who had the higher average

Display

Student 1 Display the name, scores, and average of student 1

Student 2 Display the name, scores, and average of student 2

The Code

Here is the code for the example:

import java.awt.\*;

import JavaQuickWindows.\*;

public class StudentInterface extends GBFrame{

Label student1Label = addLabel ("Student 1",1,2,1,1);

Label student2Label = addLabel ("Student 2",1,3,1,1);

Label displayLabel = addLabel ("Display" ,1,4,1,1);

Label namesLabel = addLabel ("Names" ,2,1,1,1);

Label test1Label = addLabel ("Test 1" ,3,1,1,1);

Label test2Label = addLabel ("Test 2" ,4,1,1,1);

Label test3Label = addLabel ("Test 3" ,5,1,1,1);

TextField stud1NameField = addTextField ("",2,2,1,1);

IntegerField stud1Test1Field = addIntegerField (0 ,3,2,1,1);

IntegerField stud1Test2Field = addIntegerField (0 ,4,2,1,1);

IntegerField stud1Test3Field = addIntegerField (0 ,5,2,1,1);

private TextField stud2NameField = addTextField ("",2,3,1,1);

private IntegerField stud2Test1Field = addIntegerField (0 ,3,3,1,1);

private IntegerField stud2Test2Field = addIntegerField (0 ,4,3,1,1);

private IntegerField stud2Test3Field = addIntegerField (0 ,5,3,1,1);

private TextArea displayField = addTextArea ("",2,4,2,4);

private Button storeButton = addButton("Store Data",6,1,5,1);

MenuItem highTest1MI = addMenuItem ("HighStudent", "Test1");

MenuItem highTest2MI = addMenuItem ("HighStudent", "Test2");

MenuItem highTest3MI = addMenuItem ("HighStudent", "Test3");

MenuItem highOverallMI = addMenuItem ("HighStudent", "Overall");

MenuItem highAverageMI = addMenuItem ("HighStudent", "Average");

MenuItem displayStudent1MI = addMenuItem ("Display", "Student1");

MenuItem displayStudent2MI = addMenuItem ("Display", "Student2");

private Student student1;

private Student student2;

public StudentInterface(){

student1 = new Student();

student2 = new Student();

setTitle ("Student Scores");

}

public void buttonClicked (Button buttonObj){

student1.setName (stud1NameField.getText());

student1.setScore (1, stud1Test1Field.getNumber());

student1.setScore (2, stud1Test2Field.getNumber());

student1.setScore (3, stud1Test3Field.getNumber());

student2.setName (stud2NameField.getText());

student2.setScore (1, stud2Test1Field.getNumber());

student2.setScore (2, stud2Test2Field.getNumber());

student2.setScore (3, stud2Test3Field.getNumber());

}

public void menuItemSelected (MenuItem menuItemObj){

if (menuItemObj == highTest1MI)

compareAndReport

("Test 1", student1.getScore(1), student2.getScore(1));

else if (menuItemObj == highTest2MI)

compareAndReport

("Test 2", student1.getScore(2), student2.getScore(2));

else if (menuItemObj == highTest3MI)

compareAndReport

("Test 3", student1.getScore(3), student2.getScore(3));

else if (menuItemObj == highOverallMI)

compareAndReport

("Overall", student1.getHighScore(), student2.getHighScore());

else if (menuItemObj == highAverageMI)

compareAndReport

("Average", student1.getAverage(), student2.getAverage());

else if (menuItemObj == displayStudent1MI)

displayField.setText(student1.toString());

else if (menuItemObj == displayStudent2MI)

displayField.setText("" + student2);

}

private void compareAndReport (String description, int stud1, int stud2){

String str = description + ": ";

if (stud1 == stud2)

str = str + "the students are equal";

else if (stud1 > stud2)

str = str + student1.getName() + " is higher.";

else

str = str + student2.getName() + " is higher.";

messageBox (str);

}

public static void main (String[] args){

Frame frm = new StudentInterface();

frm.setSize (400, 250);

frm.setVisible(true);

}

}