

Chapter 14

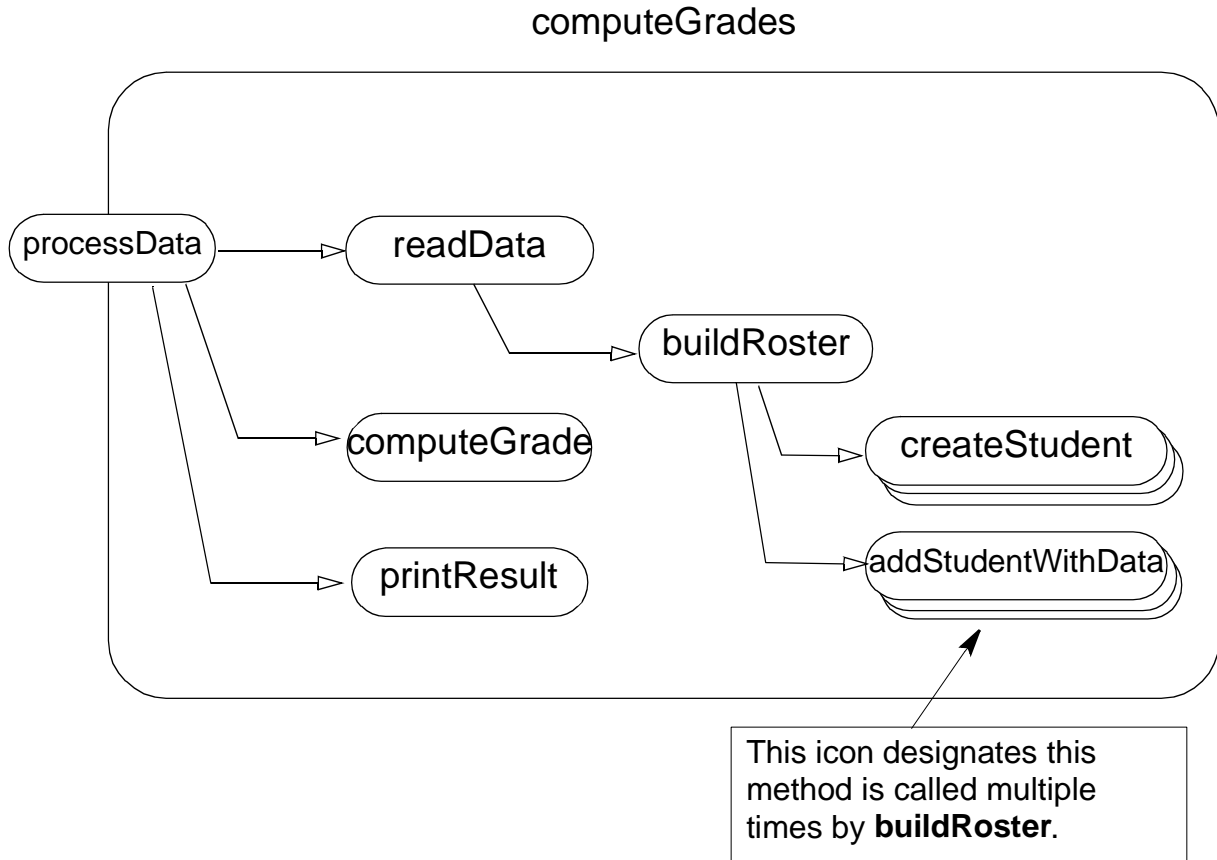
Class Roster Maintenance Program

OBJECTIVES

After you have read and studied this chapter, you should be able to

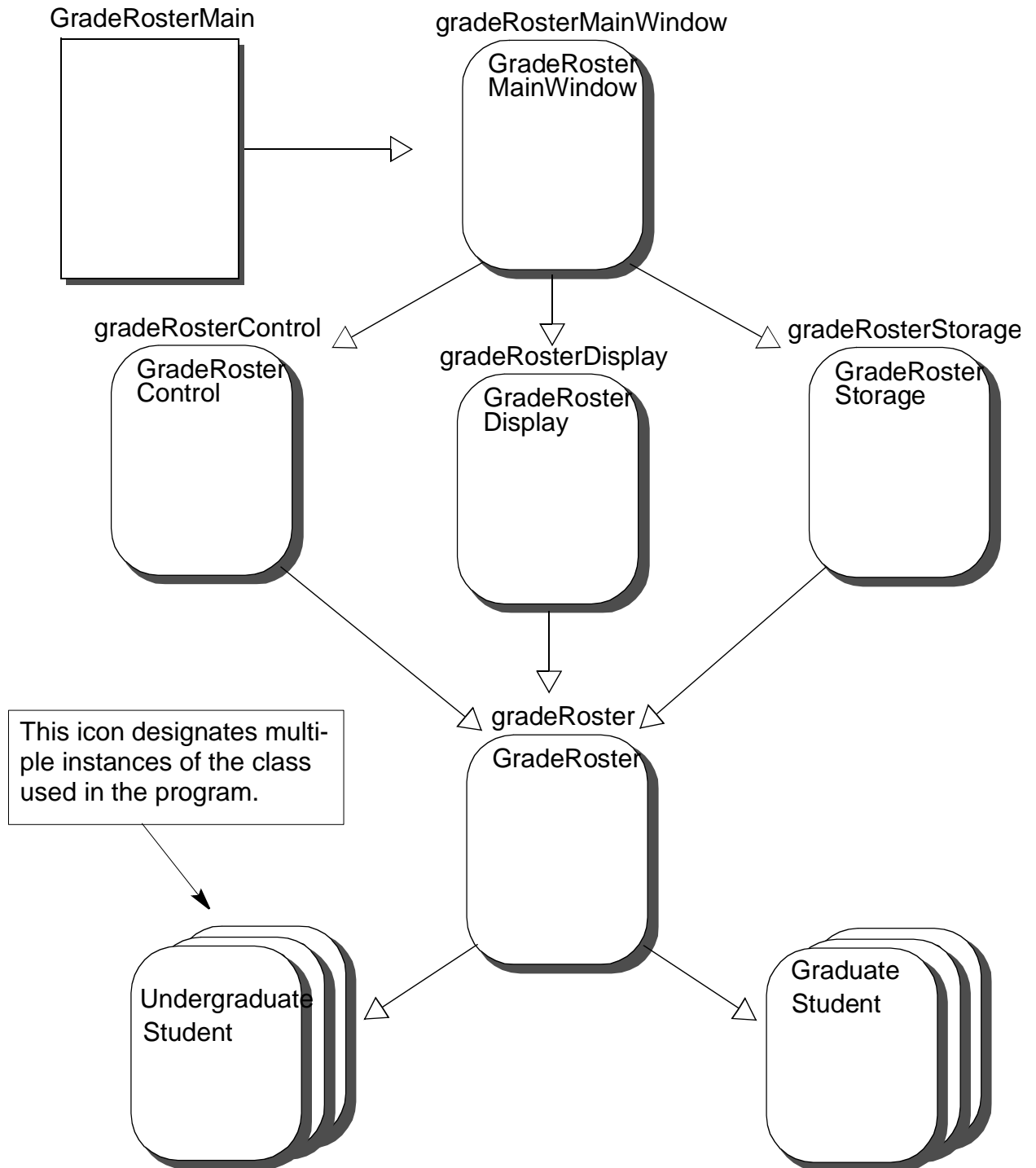
- Develop large programs incrementally using multiple objects from object categories controller, storage, application logic, and user interface.
- Develop large programs that are extensible and modifiable by applying polymorphism and inheritance effectively in program design.
- Document how the methods in the classes are related by using method call sequence diagrams.

FIGURE 14.1 Method call sequence of the **processData** method. Only the methods of **ComputeGradesMainWindow** are shown in the diagram.



	Objects	To Do
Step 1	GradeRoster Main	Implement the full class. The main method creates an instance of GradeRosterMainWindow and calls its setVisible method.
	GradeRoster MainWindow	Implement the shell of the main class that includes menu choices and corresponding methods for processing menu choices.

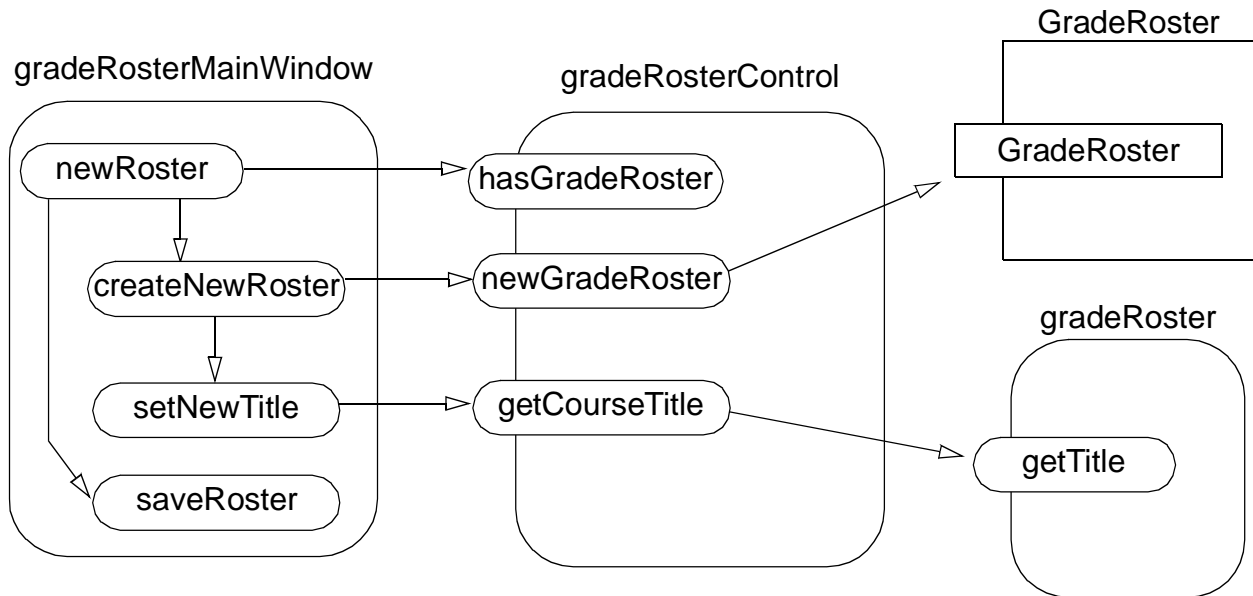
FIGURE 14.2 A simplified object diagram for the grade roster program. Primary objects are shown here. Secondary objects such as `MessageBox`, `InputDialog`, and others are not shown.



	Objects	To Do
Step 2	GradeRoster MainWindow	Implement the methods newRoster createNewRoster setNewTitle Modify the constructor as necessary. Add a ResponseBox called saveBox as a new data member. Define constants SAVE, NO_SAVE, and CANCEL.
	GradeRoster Control	Implement the methods <constructor> getCourseTitle hasGradeRoster newGradeRoster Add InputBox as a new data member.
	GradeRoster	Implement the methods <constructor> getTitle

	Objects	To Do
Step 3	GradeRoster MainWindow	Implement the method addNewStudent
	GradeRoster Control	Implement the methods addStudent addGraduateStudent addUndergraduateStudent Modify the constructor as necessary.
	GradeRoster	Implement the methods addStudent getStudent expandRoster

FIGURE 14.3 Method call sequence for the **newRoster** method. Calls to standard **javabook** objects are not shown.



	Objects	To Do
Step 4	GradeRoster MainWindow	Implement the method deleteStudent
	GradeRoster Control	Implement the method deleteStudent
	GradeRoster	Implement the methods deleteStudent compactRoster

FIGURE 14.4 Method call sequence for the **addNewStudent** method. Calls to standard **javabook** objects are not shown.

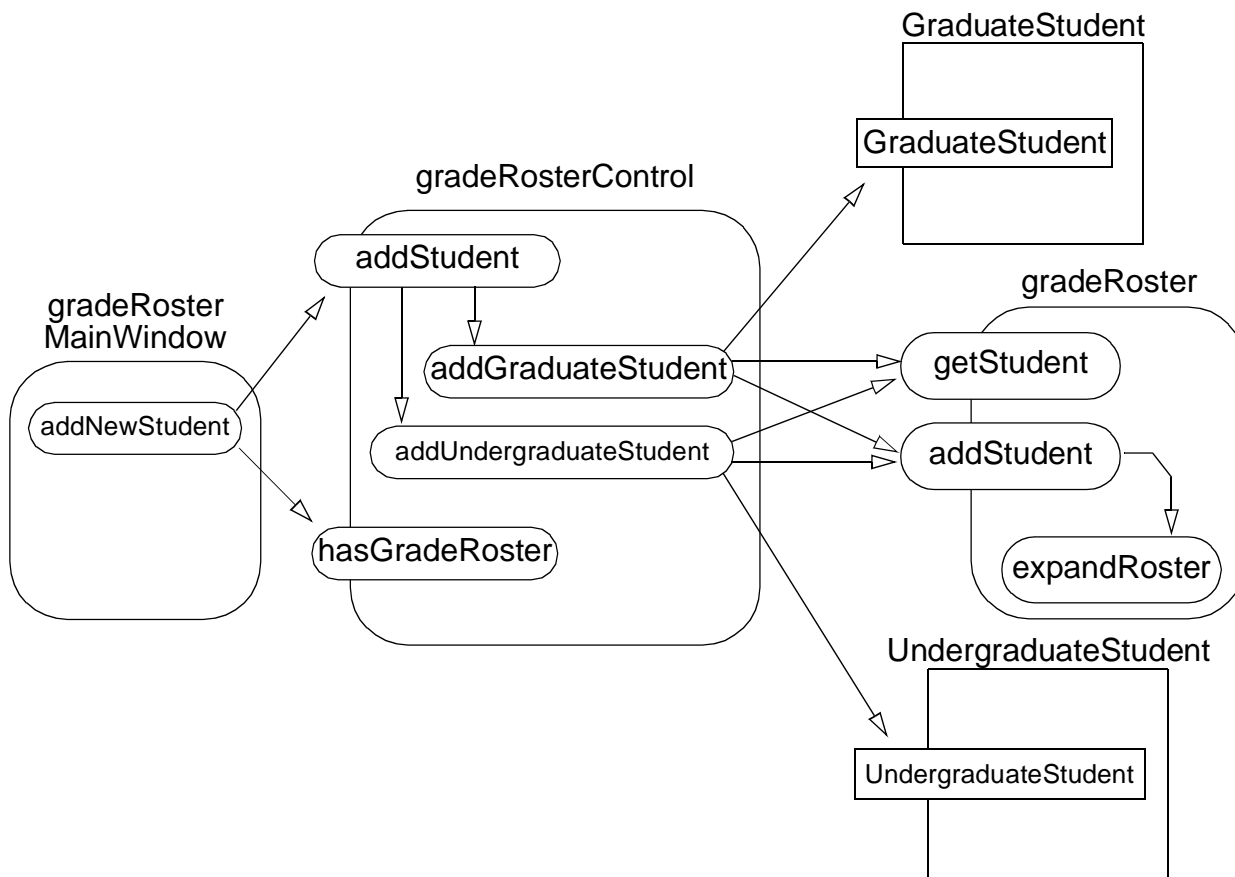


FIGURE 14.6 A **StudentNameDialog** dialog for inputting the current and new names of a student.

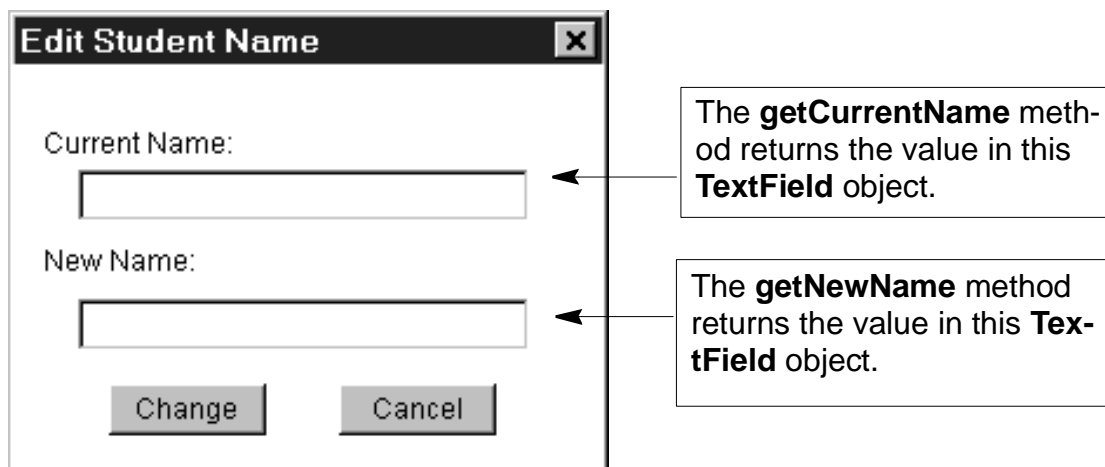
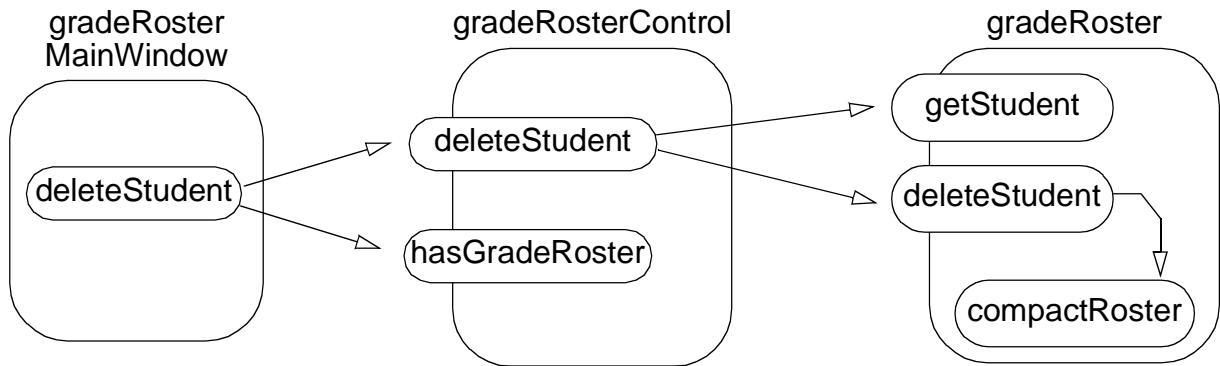


FIGURE 14.5 Method call sequence for the **deleteStudent** method. Calls to standard **javabook** objects are not shown.



	Objects	To Do
Step 5	GradeRoster MainWindow	Implement the method editStudentName
	GradeRoster Control	Implement the method editStudentName Modify the constructor and data member declara- tion as necessary.
	StudentName Dialog	Define this class fully.

FIGURE 14.7 An extended object diagram for the grade roster program now includes **StudentNameDialog**.

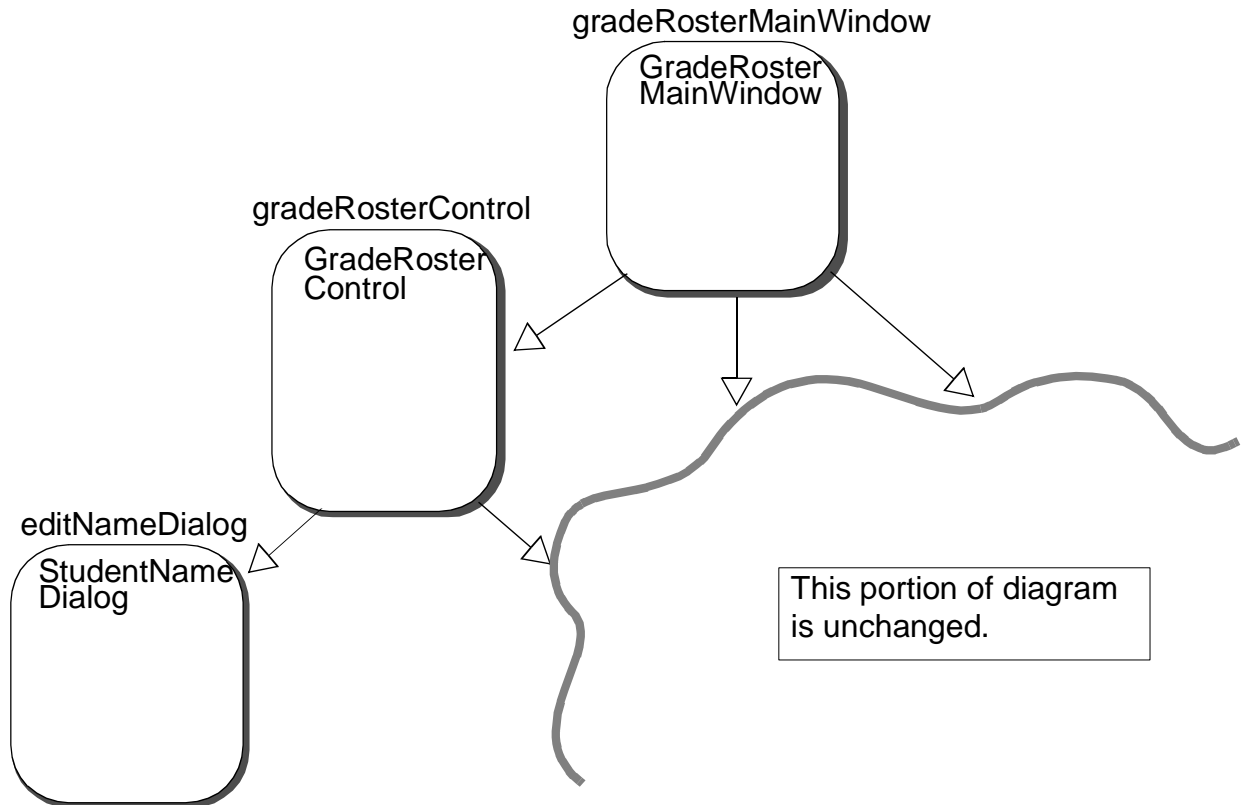


FIGURE 14.9 A sample **TestScoreDialog** object for editing test scores.

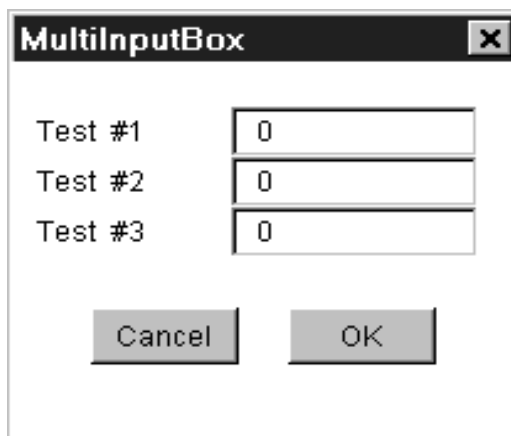


FIGURE 14.8 A method call sequence diagram for the **editStudentName** method. Calls to standard **javabook** objects are not shown.

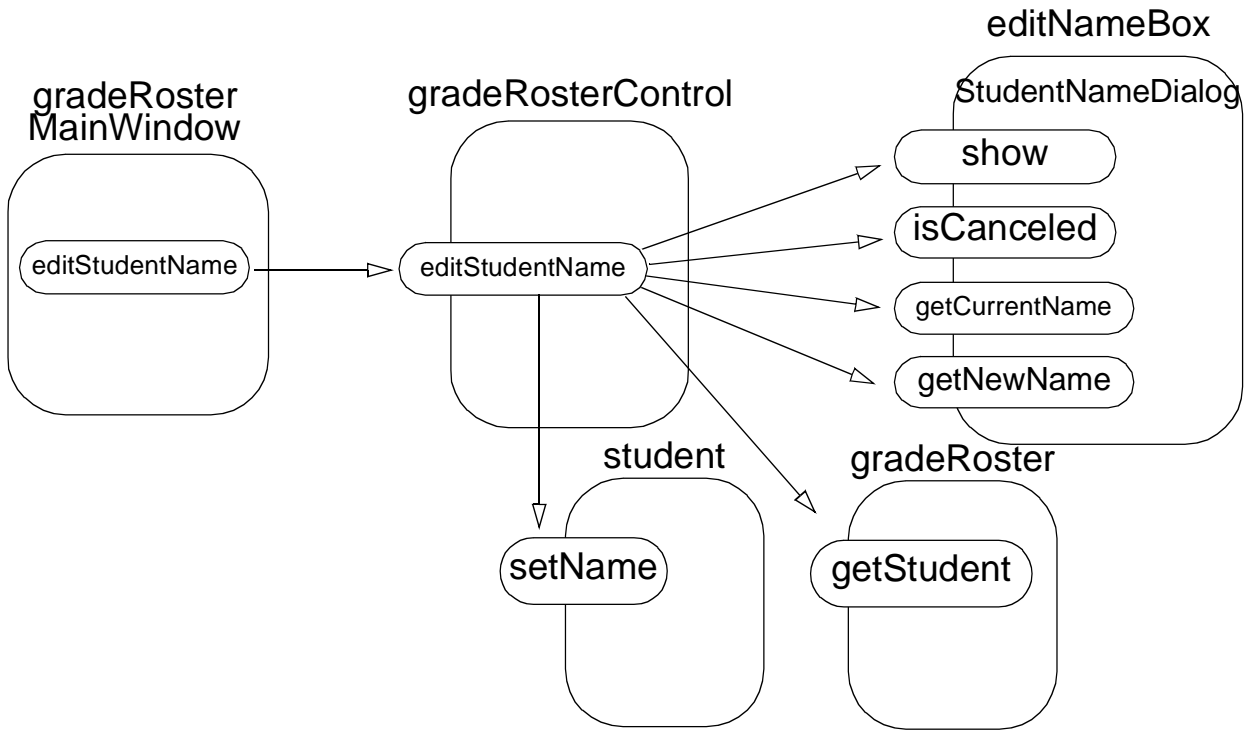
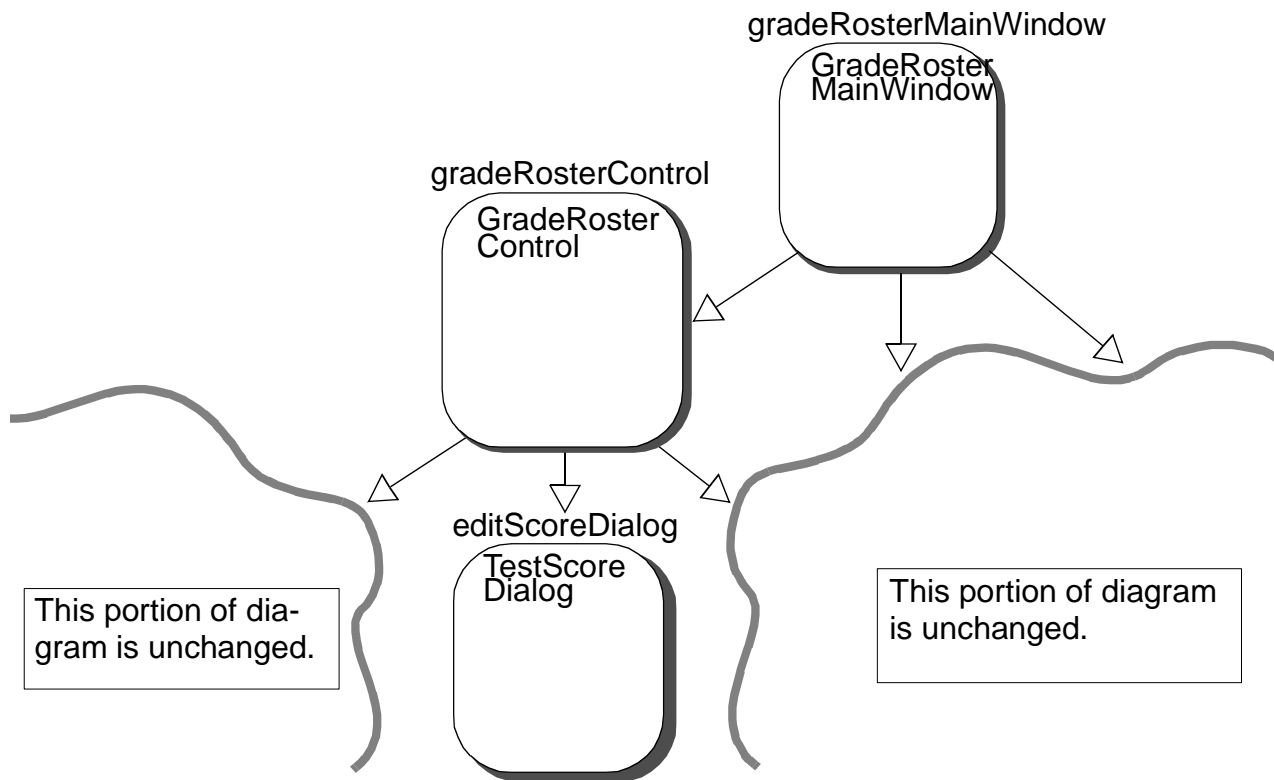
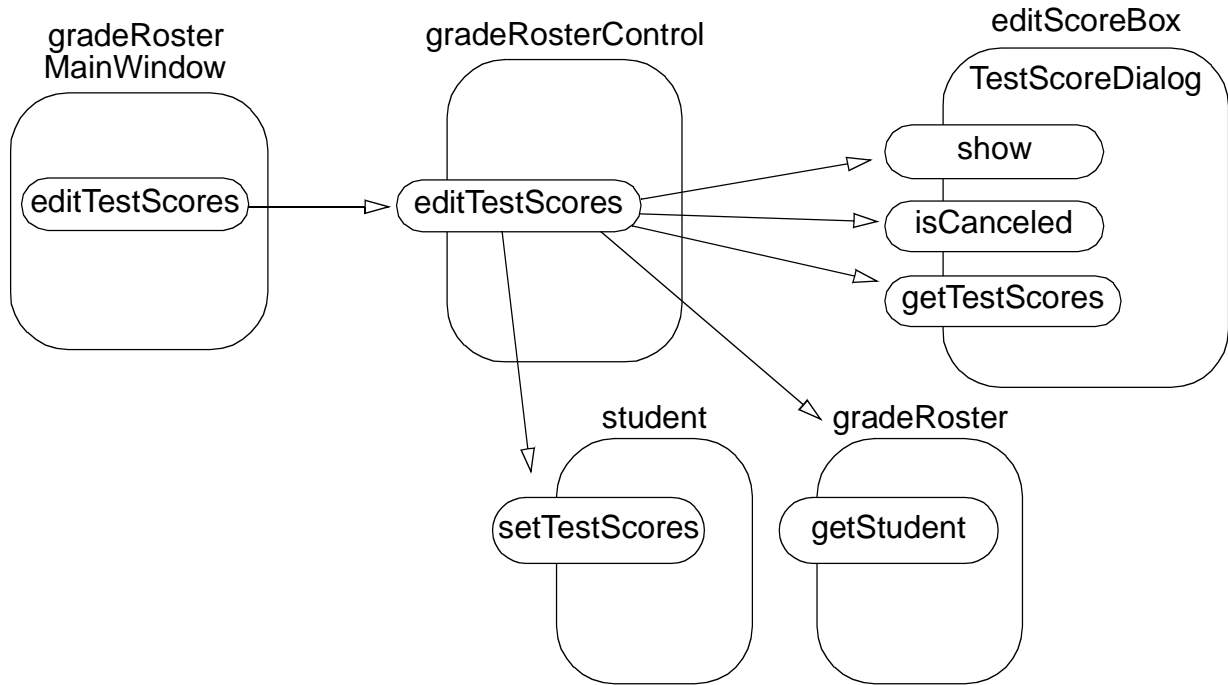


FIGURE 14.10 An extended object diagram for the grade roster program now includes **TestScoreDialog**.



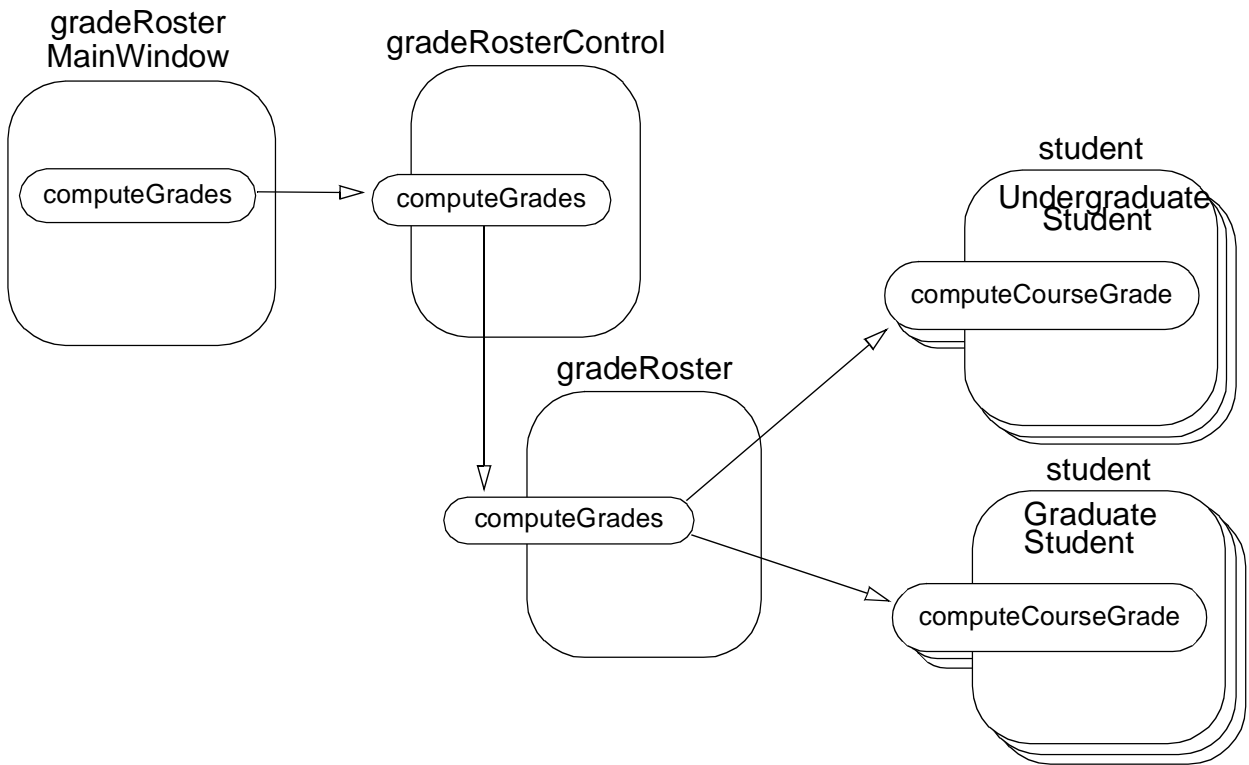
	Objects	To Do
Step 6	GradeRoster MainWindow	Implement the methods editTestScores computeGrades
	GradeRoster Control	Implement the methods editTestScores computeGrades Modify the constructor and data member declaration as necessary.
	GradeRoster	Implement the method computeGrades
	TestScore Dialog	Implement the class fully.

FIGURE 14.11 Method call sequence for the **editTestScores** method. Calls to standard **javabook** objects are not shown.



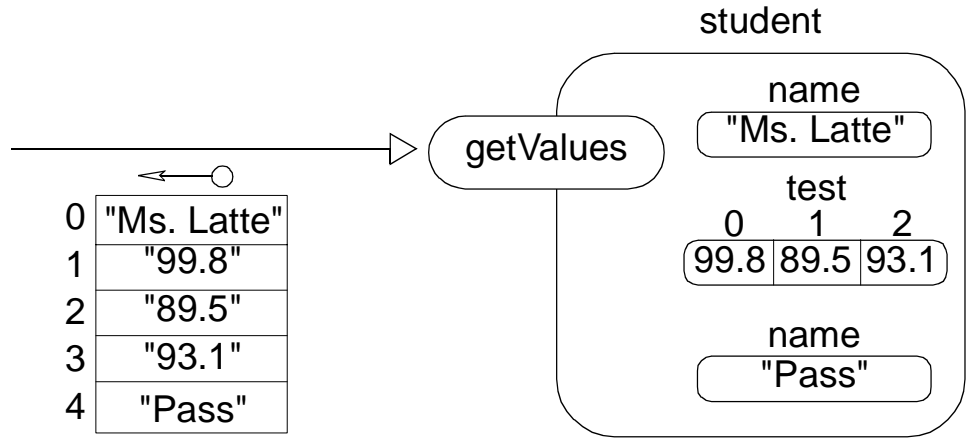
Name	Test1	Test2	Test 3	Grade
Mr. Espresso	76.3	88.4	92.2	Pass
Ms. Latte	76.3	88.4	92.2	Pass
Mr. Banacha	56.3	68.4	52.2	No Pass

FIGURE 14.12 Method call sequence for the **computeGrades** method. Calls to standard **javabook** objects are not shown.



	Objects	To Do
Step 7	GradeRoster MainWindow	Implement the methods showAllStudents showOneStudent Modify the constructor to create and initial- ize an instance of GradeRosterDisplay.
	GradeRoster Control	Implement the method getRoster
	GradeRoster Display	Implement the methods displayAll displayOne Declare the data members and the construc- tor.
	Student	Implement the method getValues
	GradeRoster	Implement the method getAllStudents

FIGURE 14.13 The **getValues** method returns an array of **String** that contains student information.



Student information is returned as an array of **String** data.

Double quotes are used here explicitly to indicate the **String** data type.

FIGURE 14.14 Method call sequence for the **showAllStudents** method. Calls to standard **javabook** objects are not shown.

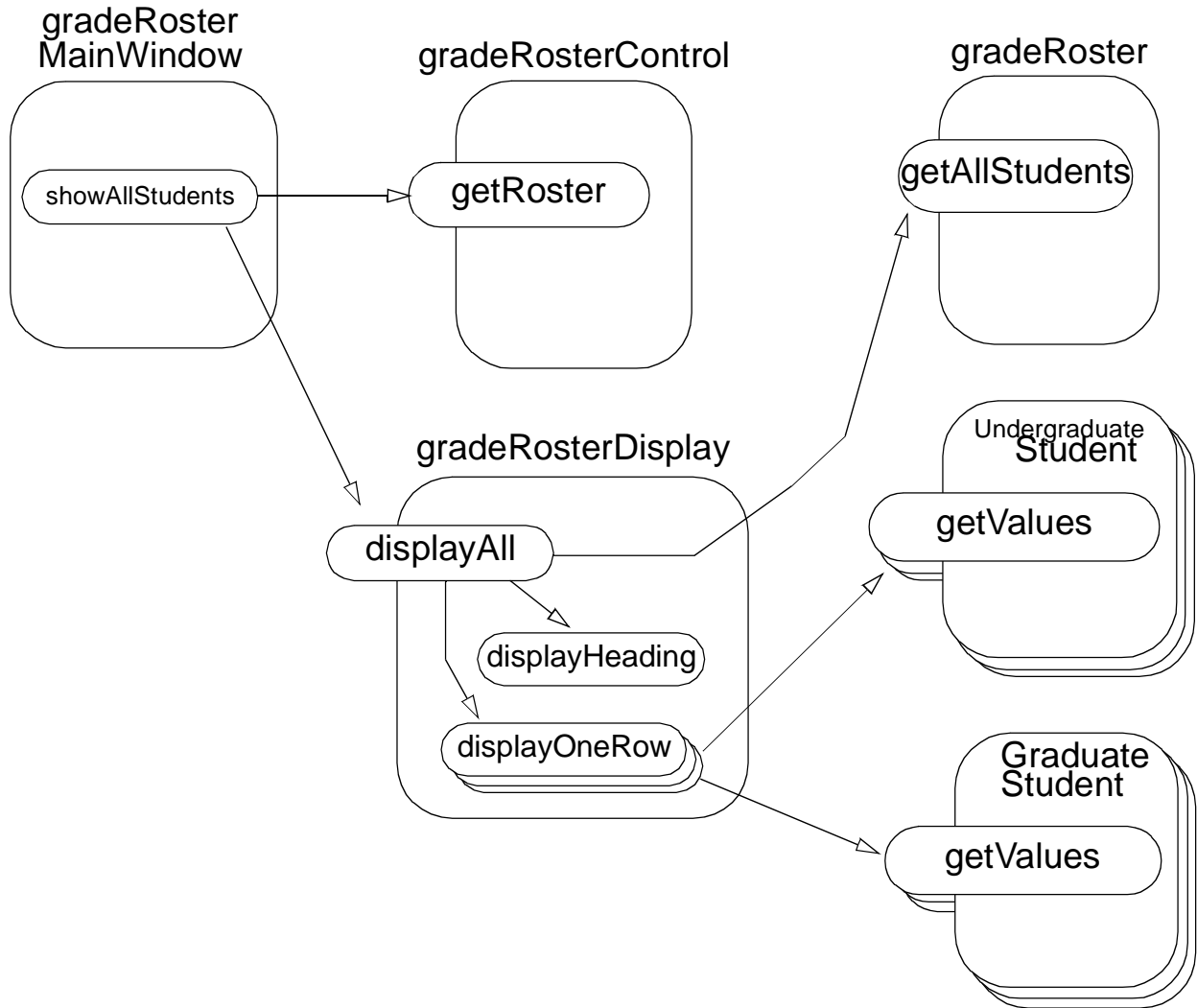
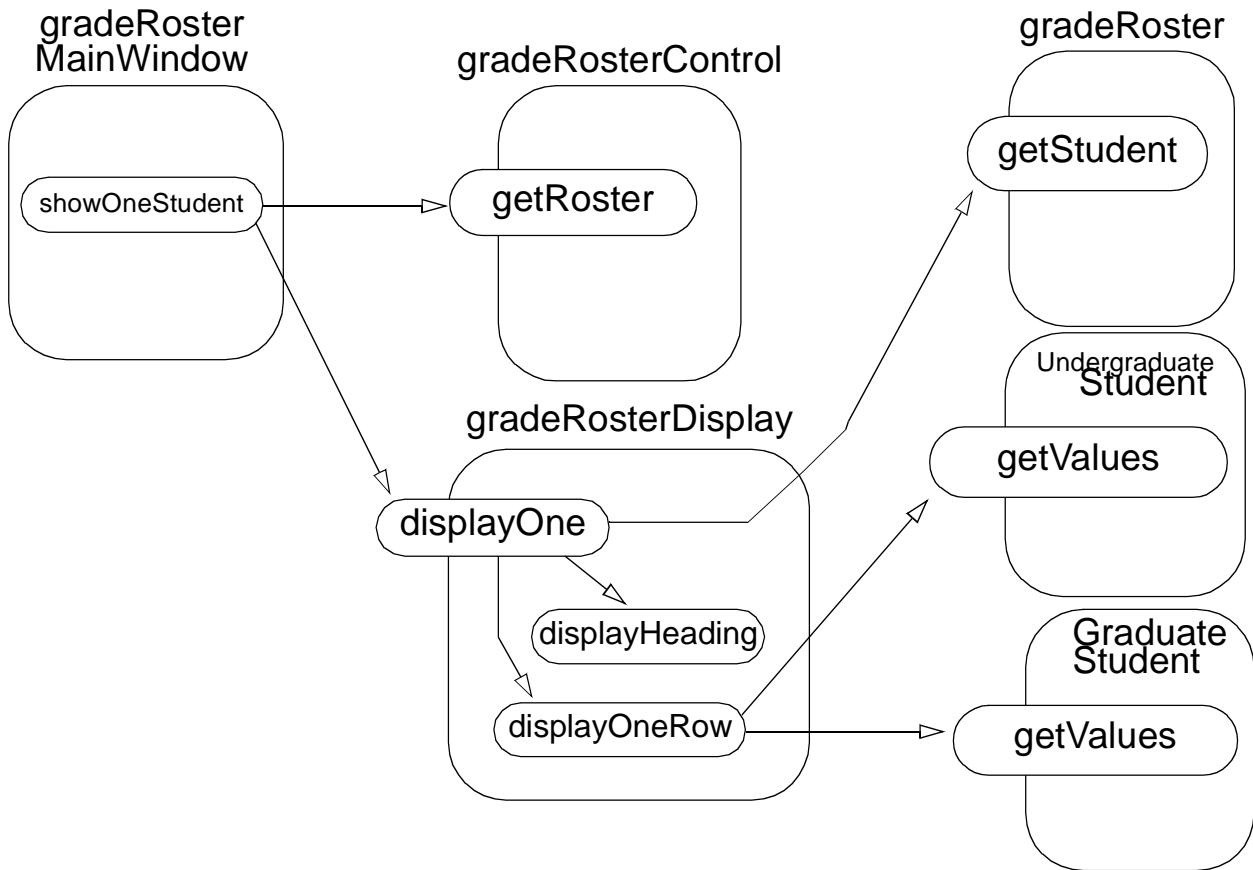


FIGURE 14.15 Method call sequence for the **showOneStudent** method. Calls to standard **javabook** objects are not shown.



	Objects	To Do
Step 8	GradeRoster MainWindow	Implement the methods openRoster saveRoster saveAsRoster Modify the constructor to create and initialize an instance of GradeRosterStorage.
	GradeRoster Control	Implement the method setGradeRoster
	GradeRoster Storage	Implement the methods save saveAs saveData open loadData Define the constructor to create and initialize data members.

FIGURE 14.16 Method call sequence for the **saveAsRoster** method. Calls to the **javabook** objects are not shown.

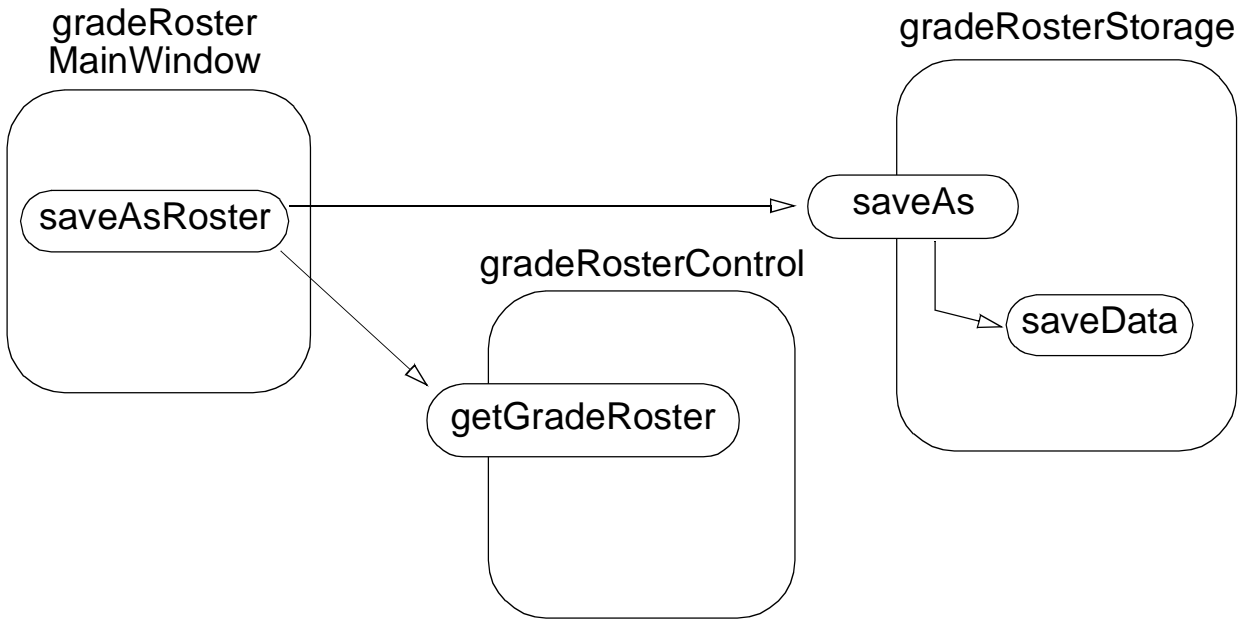


FIGURE 14.17 Method call sequence for the **saveRoster** method. Calls to the **javabook** objects are not shown.

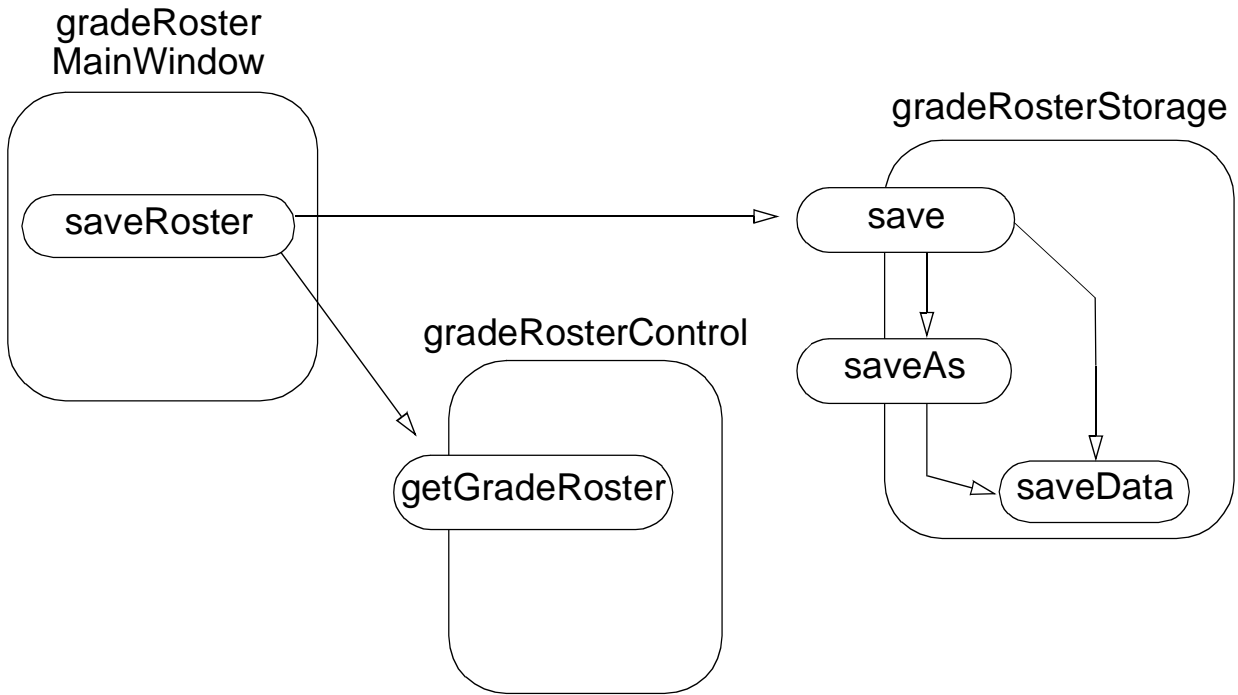


FIGURE 14.18 Method call sequence for the **openRoster** method. Calls to standard **javabook** objects are not shown.

