Topic: Simple Graphic Format Input and Output.

OOP concepts involved: Classes, Objects, Exception Handling, Static Methods, Polymorphism.

Programming generic concepts involved: Functions, Variables, Data Types, Arrays, Control Statements, Dynamic Arrays, Access Modifiers.

➢ Theoric introduction

JOptionPane Class

The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user. The JOptionPane class inherits JComponent class.

While the JOptionPane class may appear complex because of a large number of methods, almost all uses of this class are one-line calls to one of the static **showXxxDialog** methods shown below:

Method Name	Description
showConfirmDialog	Asks a confirming question, like yes/no/cancel.
showInputDialog	Prompt for some input.
showMessageDialog	Tell the user about something that has happened.
showOptionDialog	The Grand Unification of the above three.

All dialogs are modal. Each **showXxxDialog** method blocks the caller until the user's interaction is complete.

➢ Statement

Use the static methods ***showXxxDialog*** of the JOptionPane class to implement a presentation card of a college student that courses a certain College career, a certain semester and that have a number of subjects. Calculate the average grade for all the courses that the student takes.

➢ Program Code

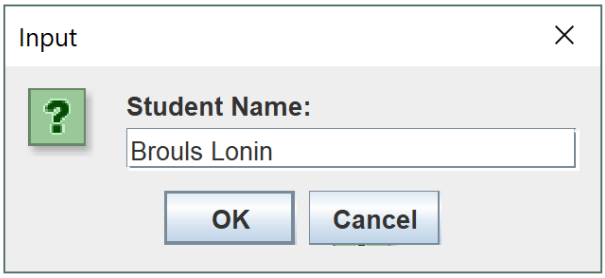
StudentPresentationCard.java

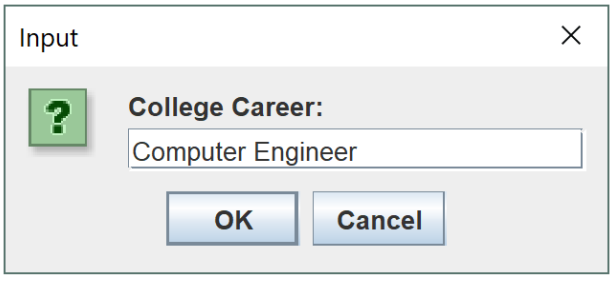
```
import java.util.ArrayList;
import javax.swing.JOptionPane;

public class StudentPresentationCard {

	public static void main(String[] args) {
		ArrayList<String> courses = new ArrayList<String>();
		ArrayList<Float> coursesGrades = new ArrayList<Float>();

		String studentInfo;
		float average = 0;

		String name = JOptionPane.showInputDialog(null, "Student Name:");
		String carrer = JOptionPane.showInputDialog(null, "College Career: ");
		String semester = JOptionPane.showInputDialog(null, "No. of Semester:
");

		int numCourses = Integer.parseInt(JOptionPane.showInputDialog(null,
"Number of courses"));

		int i = 0;
		// Saving the courses and coursesGrades of the student
		while (++i <= numCourses) {
			courses.add(JOptionPane.showInputDialog(null, "Course no."+i));

coursesGrades.add(Float.parseFloat(JOptionPane.showInputDialog(null, "Grade of
course no."+i)));
		}

		studentInfo = "Student name: "+name+"\n"+
					"College Carrer: "+carrer+"\n"+
					"Semester: "+semester+"\n\n";
```

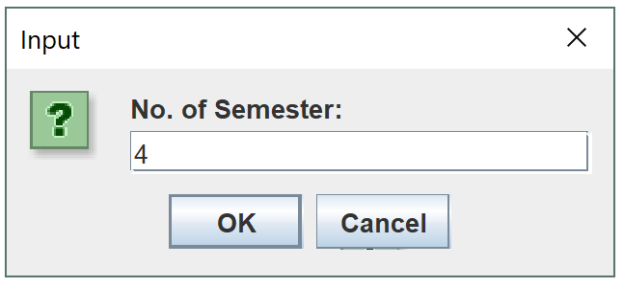
```
            for (int j=0; j<courses.size(); j++) {
                  float grade = coursesGrades.get(j);
                  average += grade;

                  studentInfo += "Course "+(j+1)+": "+courses.get(j)+"\n"+
                                       "Grade "+(j+1)+": "+grade+"\n\n";
            }

            studentInfo += "-------------------\n\n"+"Semester average: "+
(average/numCourses);
            JOptionPane.showMessageDialog(null, studentInfo, "Student Info",
JOptionPane.INFORMATION_MESSAGE);
      }

}
```

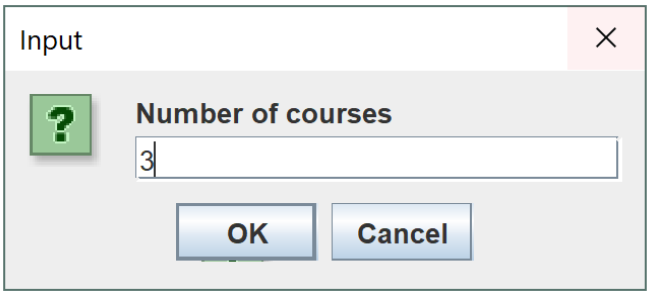
➢ Program execution

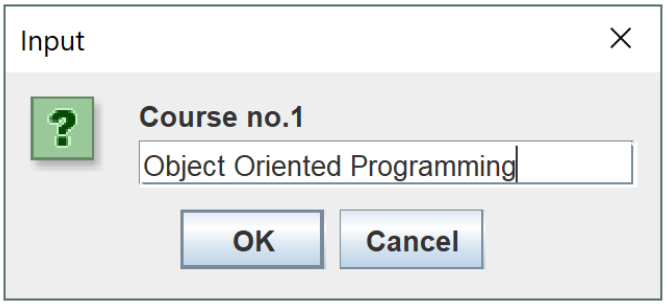
The student inserts information regarding their school.

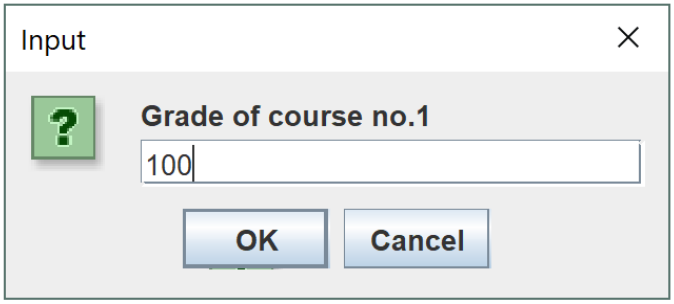




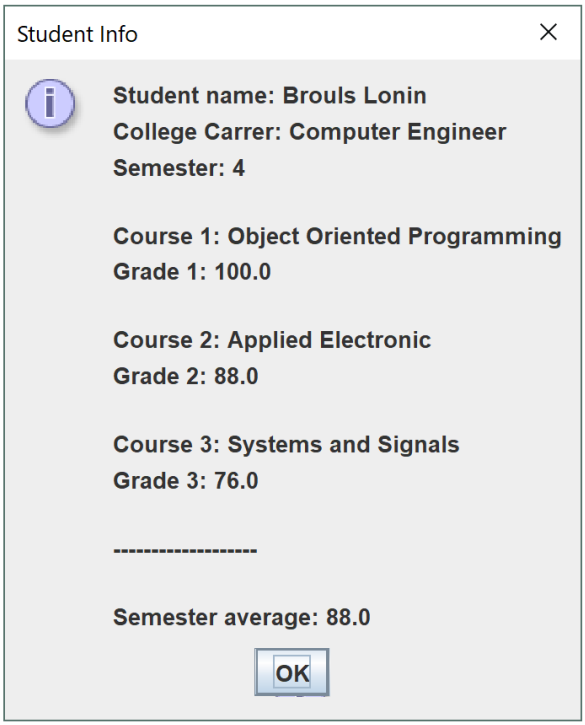








These dialogs that request the name of the course and the grade of the student will repeat 3 times because that's the number of subjects the student has.



At this moment, all the student information has been captured and an average grade of the semester has been calculated.

➢ Conclusions

The JOptionPane class is a great tool for any programmer who wants to show information graphically, as well as to capture information through dialog boxes.

With JOptionPane we have access to a large number of static methods that allow us to perform input and output operations without even creating an instance of the class.

Each method of the JOptionPane class is overloaded, so we can choose to use the one that best suits our needs.