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, 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 ask a user for an entry of their name and date of birth, and then calculate their age based on the data provided. Show the age of the person through a graphic panel.

➢ Program Code

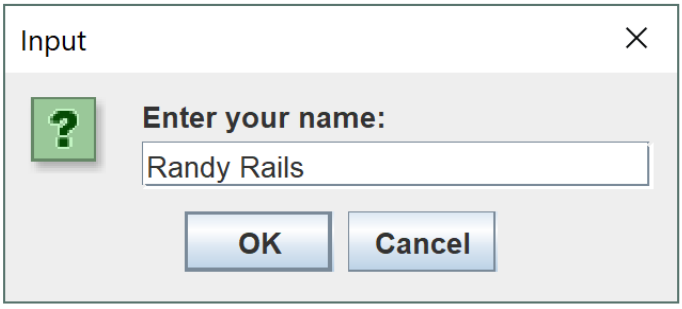
CalculateAge.java

```
import java.util.Calendar;

import javax.swing.JOptionPane;

public class CalculateAge {
	public static void main(String[] args) {
		int age, day, month, year;

		try {
			String name = JOptionPane.showInputDialog(null, "Enter your
name: ");

			String date = JOptionPane.showInputDialog(null, "Enter your
date of birth(MM/DD/YY)");

			month = Integer.parseInt(date.substring(0, 2));
			day = Integer.parseInt(date.substring(3, 5));
			year = Integer.parseInt(date.substring(6, 10));

			age = calculateAge(day, month, year);

			JOptionPane.showMessageDialog(null, name + " is " + age + "
years old!",
					"Age of " + name,
JOptionPane.INFORMATION_MESSAGE);
		} catch(Exception e) {
			JOptionPane.showMessageDialog(null, "An error has ocurred!");
		}

	}

	public static int calculateAge(int day, int month, int year) {
```

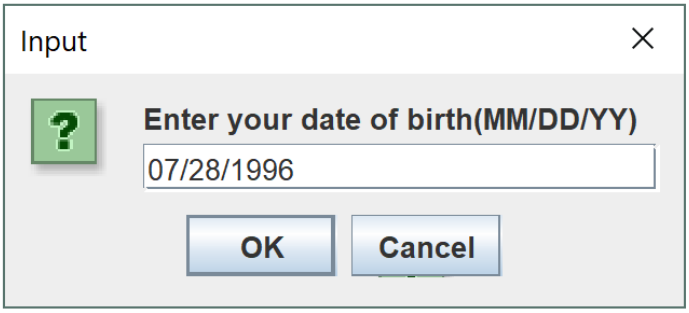
```
            int age, actualYear, actualMonth, actualDay;
            Calendar now = Calendar.getInstance();

            actualYear = now.get(Calendar.YEAR);
            actualMonth = now.get(Calendar.MONTH) + 1; // Month goes from 0 to 11
            actualDay = now.get(Calendar.DAY_OF_MONTH);

            age = actualYear - year;

            if (age > 0) {
                  if (actualMonth <= month)
                        if (actualDay < day)
                              age--;
                  return age;
            } else
                  return -1; // Not a right year
      }
}
```

➢ **Program execution**

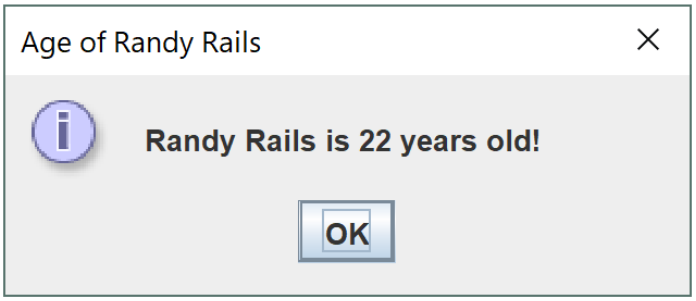
The user enters their name that is saved in the name String variable.



The user enters their date of birth in a format MM/DD/YYYY that is stored in the date variable.



The user's age is calculated and displayed through a graphic panel with their respective name.



➢ 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.