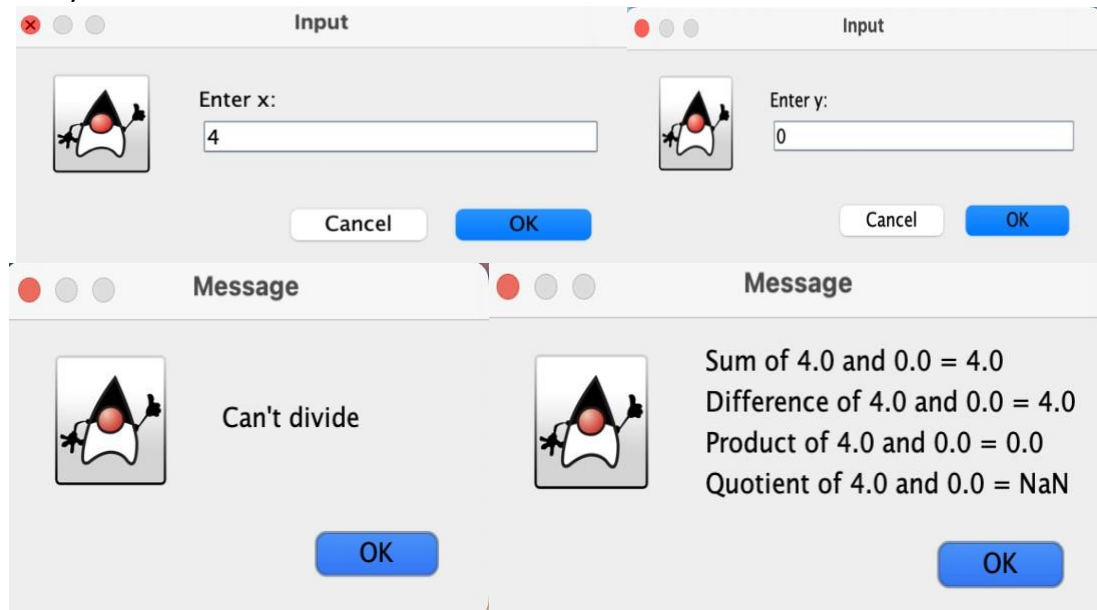


EX 2.2.5 Write a program to calculate sum, difference, product, and quotient of 2 double numbers which are entered by users.

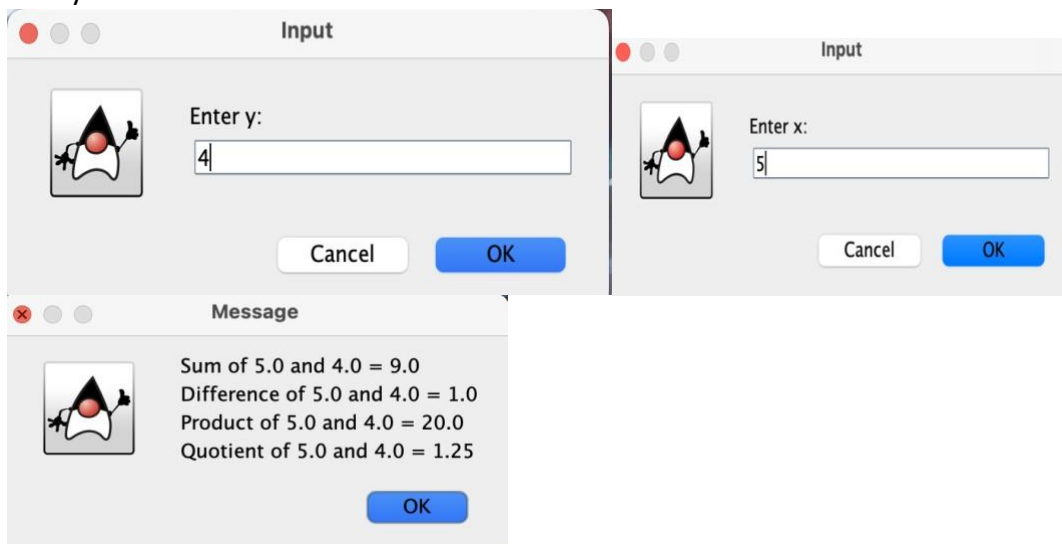
Notes

- To convert from String to double, you can use
double num1 = Double.parseDouble(strNum1)
- Check the divisor of the division

Case y = 0



case y # 0



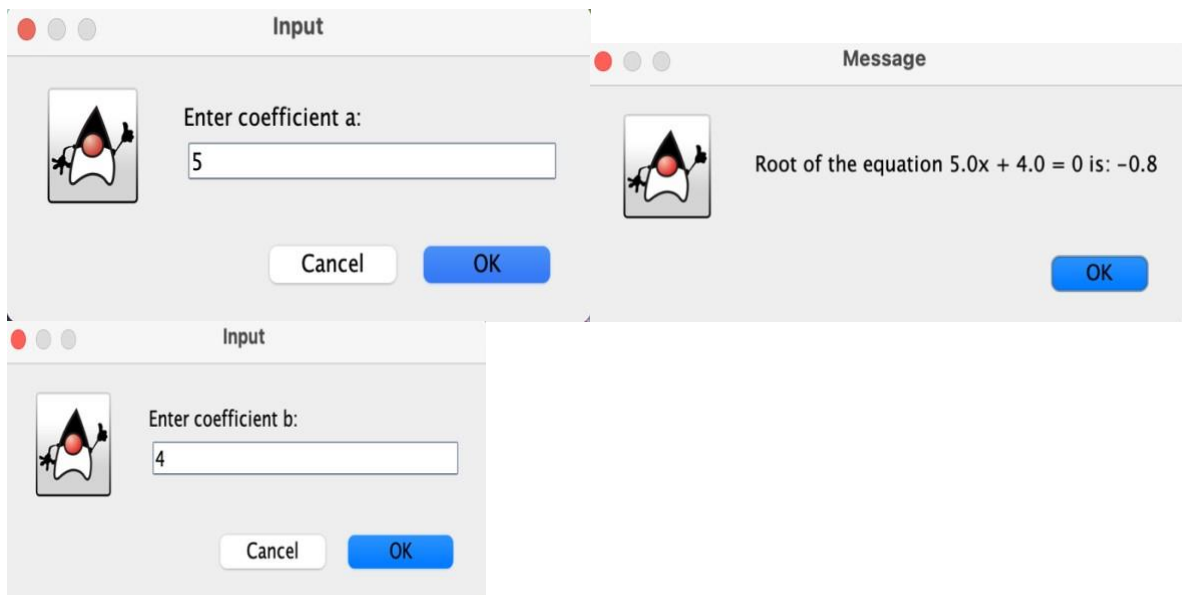
Ex 2.2.6 -Write a program to solve:

- **The first-degree equation (linear equation) with one variable**

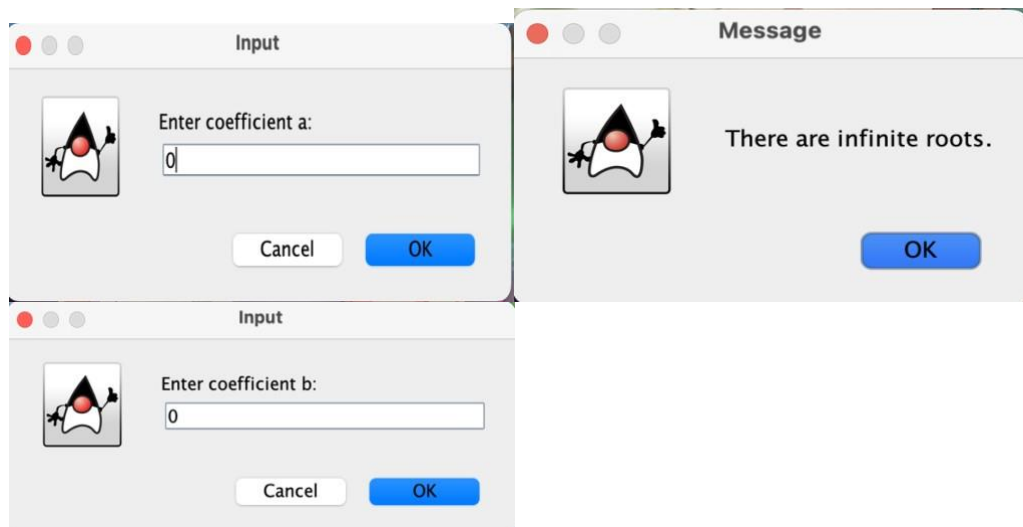
Note: A first-degree equation with one variable can have a form such as $ax+b=0$ ($a \neq 0$) $ax+b=0$ $a \neq 0$

You should handle the case where the user input value 0 for a.

- Case: $a \neq 0$



Case: $a = 0$ $b = 0$



Case $a=0, b \neq 0$



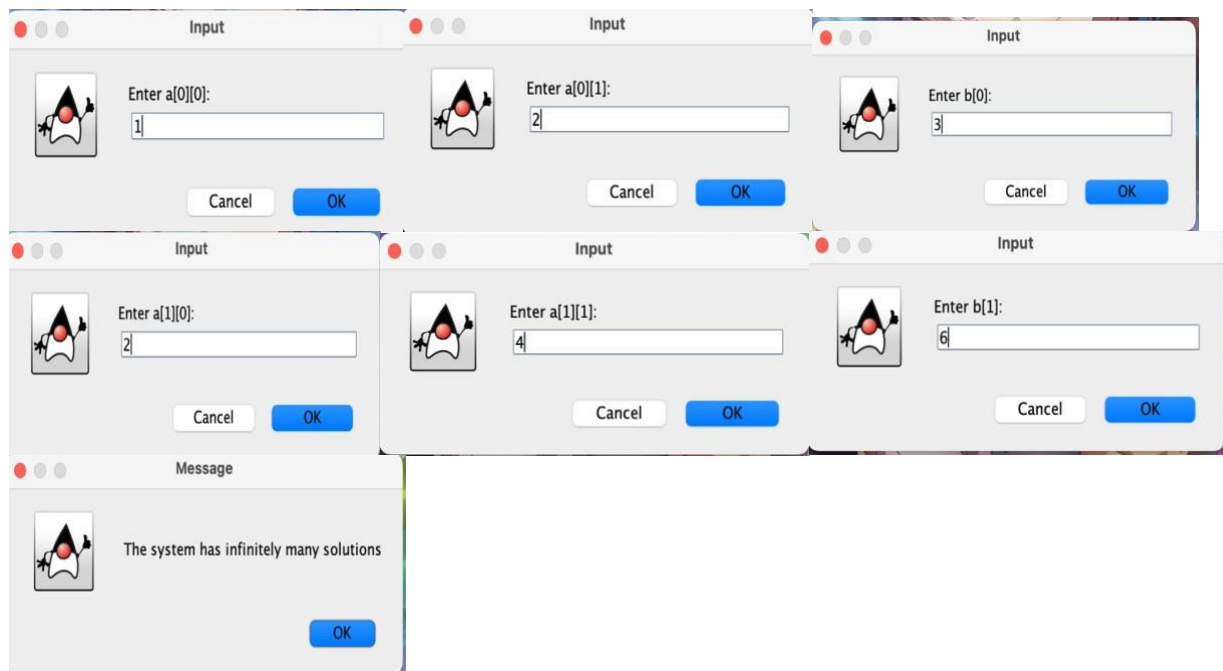
2.2.6 -The system of first-degree equations (linear system) with two variables

- **Note:** A system of first-degree equations with two variables x_1 and x_2 can be written as follows.

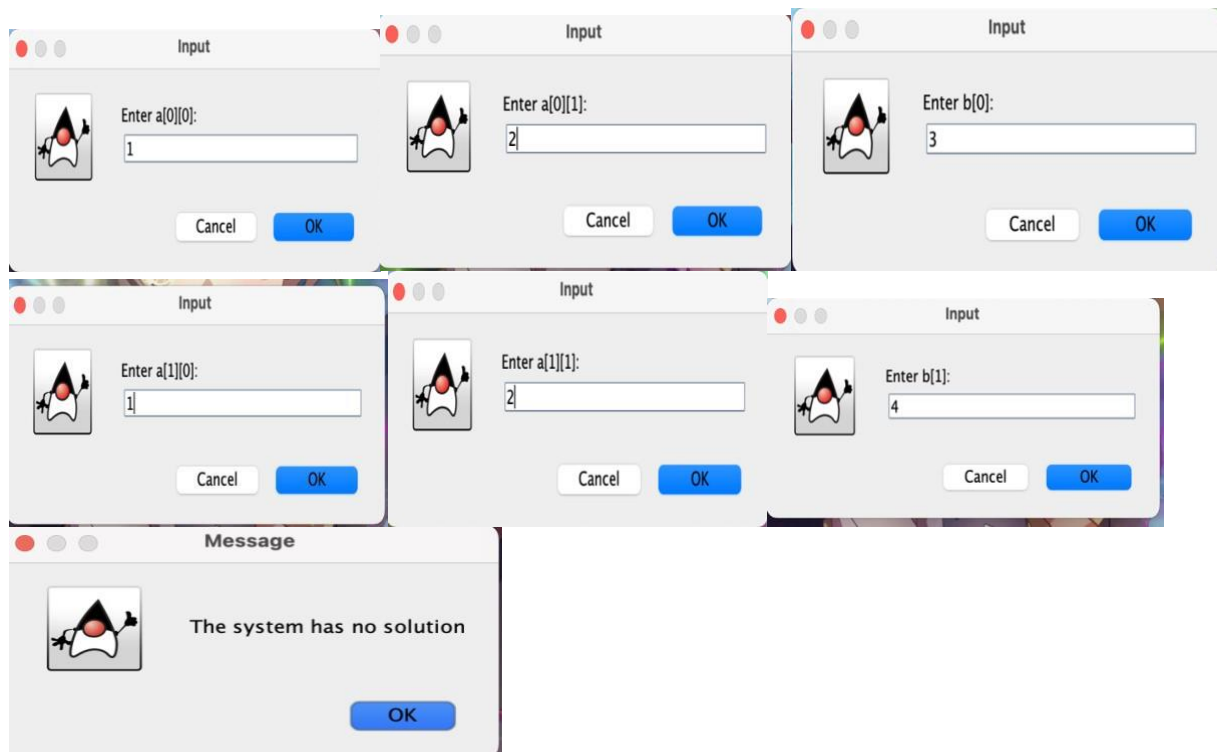
$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

You should handle the case where the values of the coefficients produce infinitely many solutions and the case where they produce no solution.

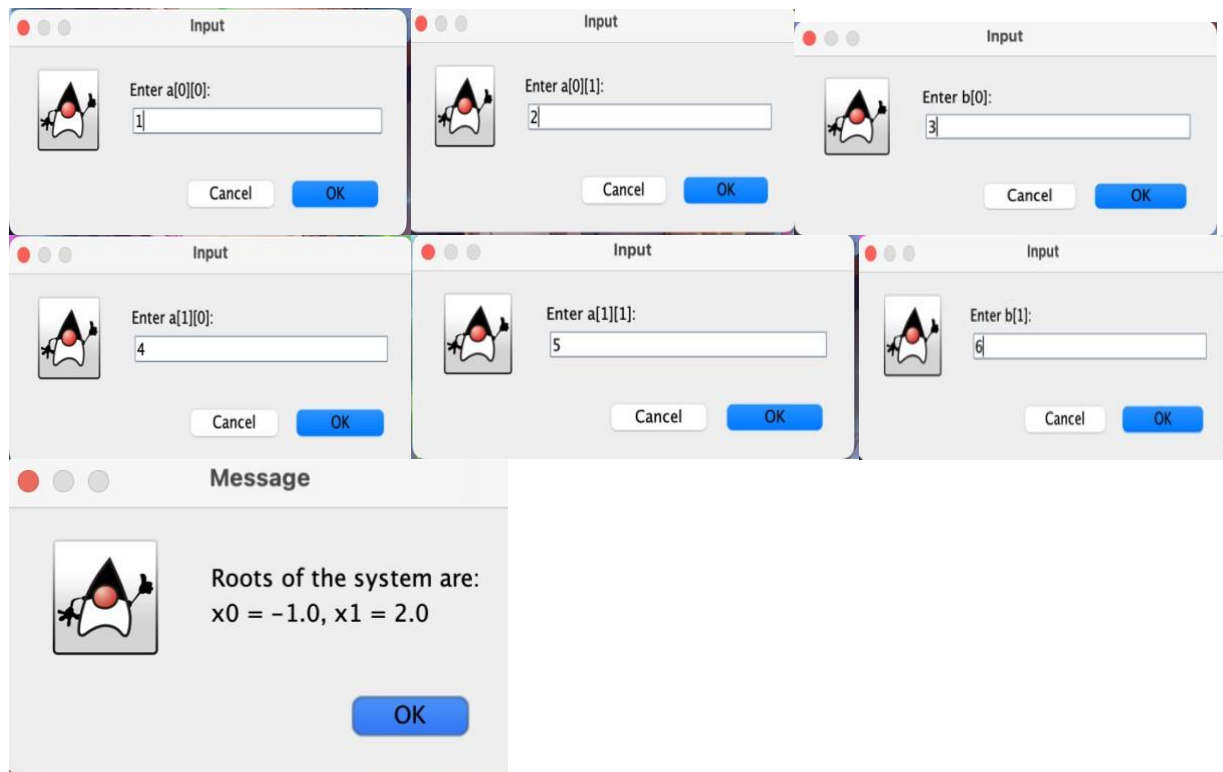
Case: infinite solution



Case: No solution



Case normal

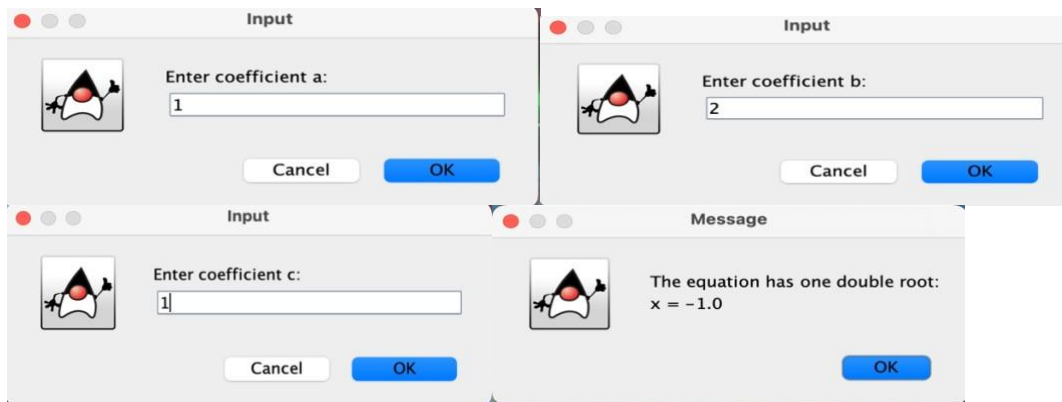


2.2.6 Second-degree

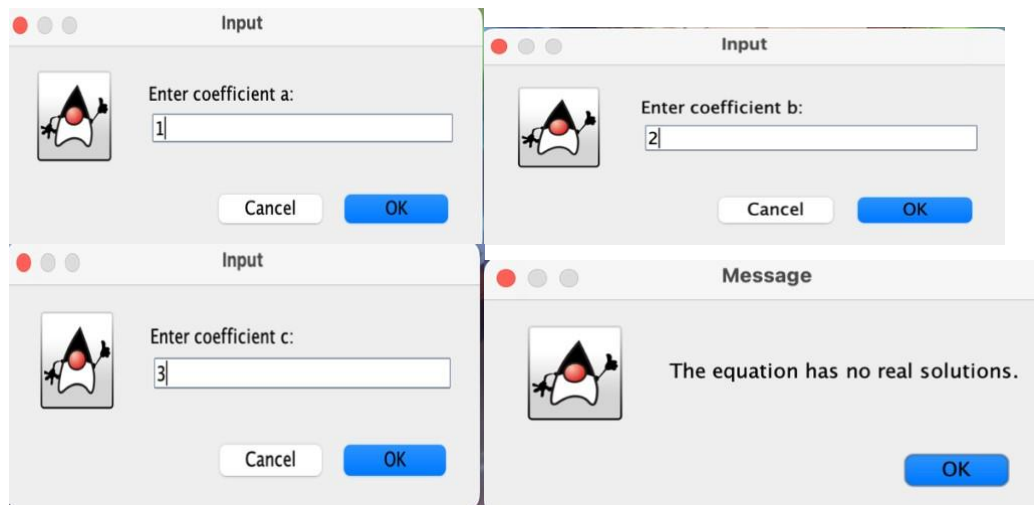
Note: A second-degree equation with one variable (i.e., quadratic equation) can have a form such as $ax^2+bx+c=0$, where x is the variable, and a , b , and c are coefficients ($a \neq 0$).

You should handle the case where the values of the coefficients produce a double root & the case where they produce no root. You should also handle the case where the user input value 0 for a .

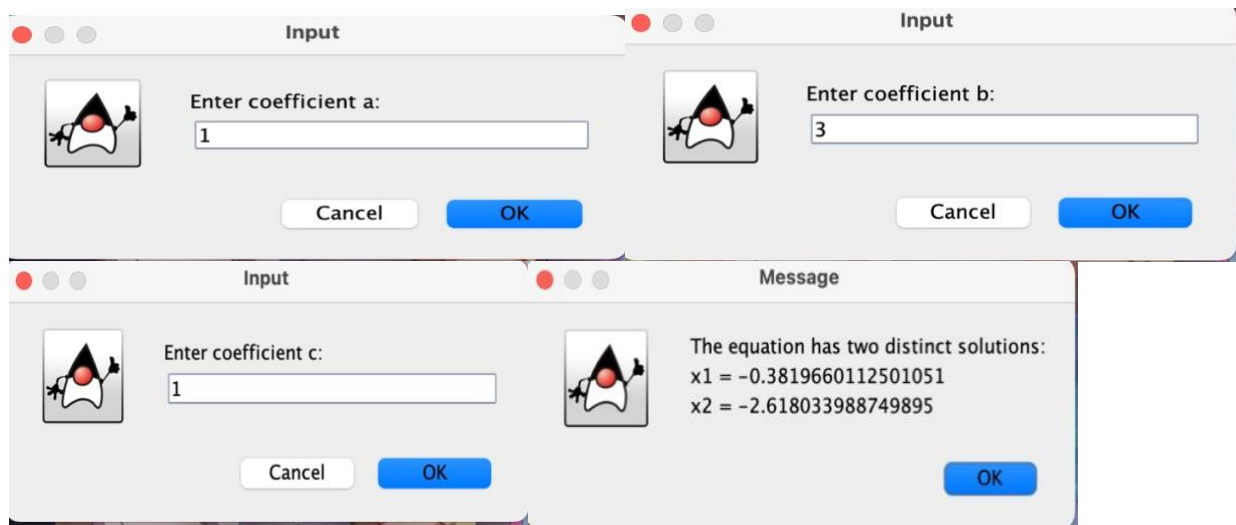
Case: Double root



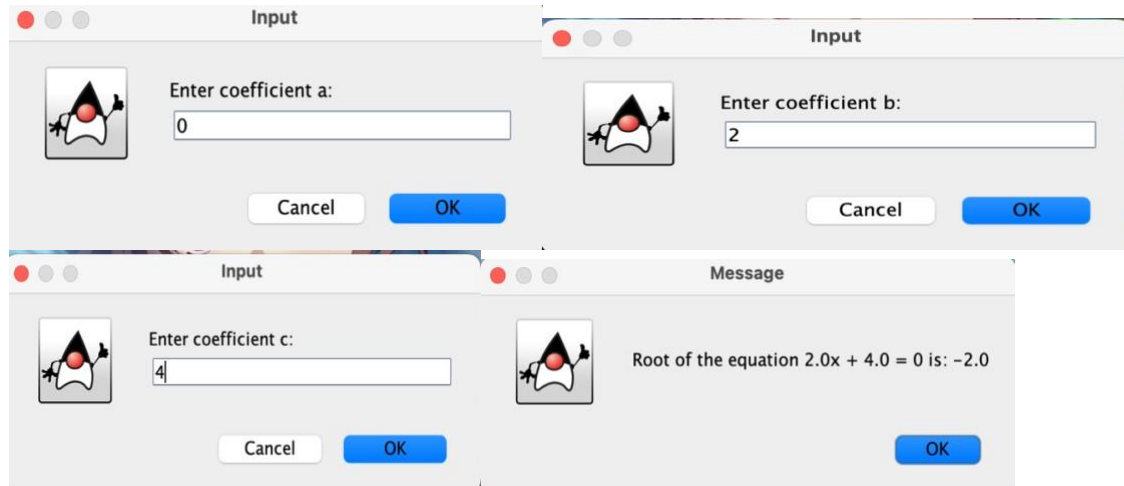
Case: No solution



Case: Normal
Case a # 0

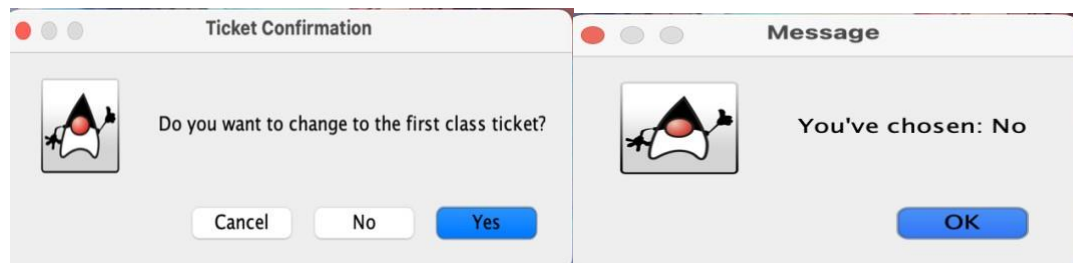


Case a=0 -> it will inherit linear model above



Exercise6.1

Question 1 What happens if users choose "Cancel"?



```

        JOptionPane.YES_NO_CANCEL_OPTION
    );
}

public void run() {
    int option = showDialog();
    JOptionPane.showMessageDialog(parentComponent:null,
        "You've chosen: " + (option == JOptionPane.YES_OPTION ? "Yes" : "No"));
}

```

Q1: If we choose cancel, It will return No. Because:

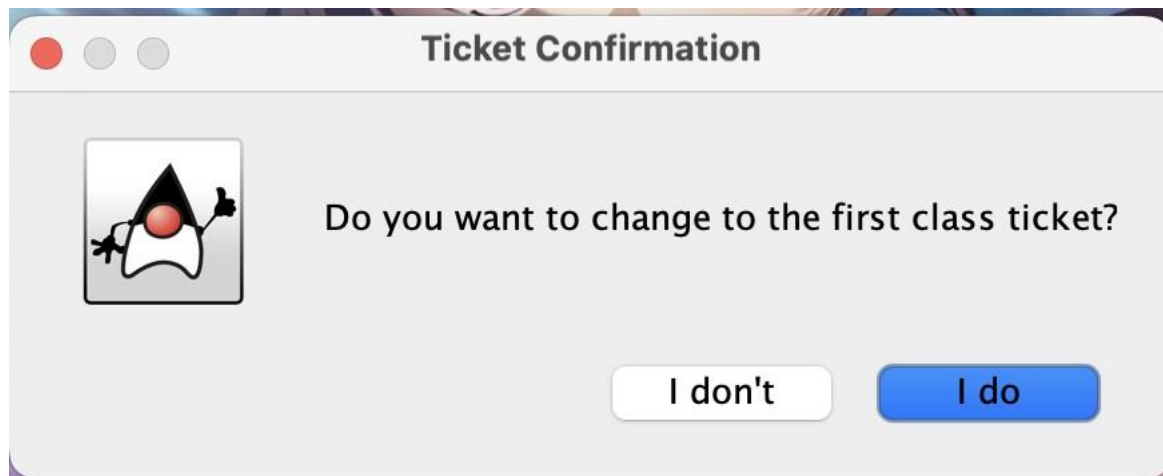
- Yes -> 0
- No -> 1
- > Cancel -> 2

Therefore, when you choose "Cancel" -> option = 2.

And the statement: option == Yes_Option -> 2 == 0

This statement is false → It returns No

Question 2: How to customize the options to users, e.g. only two options: "Yes" and "No", OR "I do" and "I don't" (Suggestion: Use Javadocs or using Eclipse/Netbean IDE help).



- Before

```
return JOptionPane.showConfirmDialog(  
    parentComponent:null,  
    message:"Do you want to change to the first class ticket?",  
    title:"Ticket Confirmation",  
    JOptionPane.YES_NO_CANCEL_OPTION  
);
```

- After

```
Object[] options = {"I do", "I don't"};  
return JOptionPane.showOptionDialog(  
    parentComponent:null,  
    message:"Do you want to change to the first class ticket?",  
    title:"Ticket Confirmation",  
    JOptionPane.YES_NO_OPTION,  
    JOptionPane.QUESTION_MESSAGE,  
    icon:null,  
    options,  
    options[0]);
```

Q2: We can customize button by creating an array "options" including { "I do", "I don't"} , and then pass "options" to the parameter of `showOptionDialog` function as we do in the above code.

options[0] -> "I do" is default -> Button "I do" is blue.

Exercise 6.2 Write a program for input/output from keyboard

```
(rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_2.java
What's your name ?
Nam
How old are you ?
18
How tall are you ?
1,65
Mr/Ms Nam is 18. Your height is 1.65
(rl-for-topic-models) apple@Fushi exercise_6 %
```

Exercise 6.3 Write a program to display a triangle with a height of n stars (*), n is entered by users.

E.g. $n=5$:

```

  *
 ***
*****
*****
*****
*****
```

```
(rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_3.java
5
  *
 ***
*****
*****
*****
*****
(rl-for-topic-models) apple@Fushi exercise_6 %
```

Exercise 6.4 Write a program to display the number of days of a month, which is entered by users (both month and year). If it is an invalid month/year, ask the user to enter again.

```
• (rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_4.java
Enter month: 0
Invalid month. Try again
Enter month: nam
Invalid month. Try again
Enter month: 2
Enter year: 2k
Invalid year. Try again
Enter year: 2100
The Month 2 of year 2100 has 28 days.
❖ (rl-for-topic-models) apple@Fushi exercise_6 %
```

Exercise 6.5 Write a Java program to sort a numeric array, and calculate the sum and average value of array elements.

```
• (rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_5.java
Enter the number of elements
4
3 56 21 910
Sorted array:
3 21 56 910 %
❖ (rl-for-topic-models) apple@Fushi exercise_6 %
```

Using quick_sort

```

5 > static void swap(int a[], int i, int j){-
10 public static int partition(int left, int right, int a[]){
11     int i = left;
12     int j = right + 1;
13     int pivot = a[left];
14     while (true) {
15         i = i + 1;
16         j = j - 1;
17         while (i <= right && a[i] < pivot){
18             i++;
19         }
20         while (j > left && pivot < a[j]){
21             j = j - 1;
22         }
23         if (i > j) break;
24         swap(a, i, j);
25     }
26     swap(a, left, j);
27     return j;
28 }
29 public static void quick_sort (int left, int right, int a []){
30     if (left < right){
31         int pivot_index = partition(left, right, a);
32         quick_sort(left, pivot_index - 1, a);
33         quick_sort(pivot_index + 1, right, a);
34     }
35 }
36
37 Run | Debug
38 public static void main(String[] args){
39     Scanner sc = new Scanner(System.in);
40     System.out.println(x:"Enter the number of elements");
41     int n = sc.nextInt();
42
43     int a[] = new int[n];
44     for (int i = 0; i < n; i++) {
45         a[i] = sc.nextInt();
46     }
47     quick_sort(left:0, n-1, a);
48     System.out.println(x:"Sorted array:");
49     for (int i = 0; i < n; i++) {
50         System.out.print(a[i] + " ");
51     }
52 }
53 }

```

Exercise 6.6 Write a Java program to add two matrices of the same size.

```

● (rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_6.java
Enter size of column
3
Enter size of row
3
Enter matrix A
1 2 3
4 5 6
7 8 9
Enter matrix B
11 12 13
14 15 16
17 18 19

The final matrix:
12.0 14.0 16.0
18.0 20.0 22.0
24.0 26.0 28.0

```