

**Universidad Tecnológica de Panamá**  
**Facultad de Ingeniería en Sistemas Computacionales**  
**Asignatura: Desarrollo Software III**  
**Ejercicio Práctico 1**

Profesor: Napoleón Ibarra

Valor: 100 puntos

Nombre: Jorge Javier Jiménez Ruiz

Cédula: 4-826-874

**Procedimiento:**

- ✓ De manera individual, desarrolle los problemas.
- ✓ Se debe entregar al profesor: Documento digital: entrega en la plataforma (TEAM) el y/o los códigos desarrollando los problemas. Sustente su trabajo en el aula de clases.

**Criterios de Evaluación:**

Criterios	Puntos (Mínimo=1, Máximo=5)	Porcentaje
Desarrollo	1-5	70 %
Sustentación	1-5	15 %
Puntualidad	1-5	15 %

**I Parte. Desarrollo de problemas en JAVA. Valor 45 Puntos**

1. Escriba un programa que lea la hora en notación de 24 horas y que imprima en notación de 12; por ejemplo, si la entrada es 13:45, la salida será 1:45 pm. El programa debe solicitar al usuario que introduzca exactamente cinco caracteres para especificar una hora. *Valor 15 puntos*
2. Escribir un programa que lea una cadena de hasta N caracteres que represente un número romano e imprima el formato de ese número y su equivalente en numeración arábica; los caracteres romanos y su equivalente son. *Valor 15 puntos*

I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Prueba de escritorio:

**LXXXVI (86), CCCXIX (319) y MCCLIV (1254).**

3. Calculadora de límite de crédito. Desarrolle una aplicación que determine si alguno de los clientes de una tienda de departamentos se ha excedido del límite de crédito en una cuenta. Para cada cliente se tienen los siguientes datos. **Valor 15 puntos**
  - a) El número de cuenta.
  - b) El saldo al inicio del mes.
  - c) El total de todos los artículos cargados por el cliente en el mes.
  - d) El total de todos los créditos aplicados a la cuenta del cliente en el mes.
  - e) El límite de crédito permitido.

El programa debe recibir como entrada cada uno de estos datos en forma de números, debe calcular el nuevo saldo (=saldo inicial + cargos - créditos), mostrar el nuevo saldo y determinar si éste excede el límite de crédito del cliente. Para los clientes cuyo límite de crédito sea excedido, el programa debe mostrar el mensaje “Se excede el límite de su crédito”.

4. Encontrar el número más grande. El proceso de encontrar el valor más grande se utiliza con frecuencia en aplicaciones de computadora. **Valor 15 puntos**

Por ejemplo, un programa para determinar el ganador de un concurso de ventas recibe como entrada el número de unidades vendidas por cada vendedor. El vendedor de haya vendido más unidades es el que gana el concurso. Escriba un programa que reciba como entrada una serie de 10 números y que determine e imprima el mayor de los números. Su programa debe utilizar cuando menos las siguientes variables:

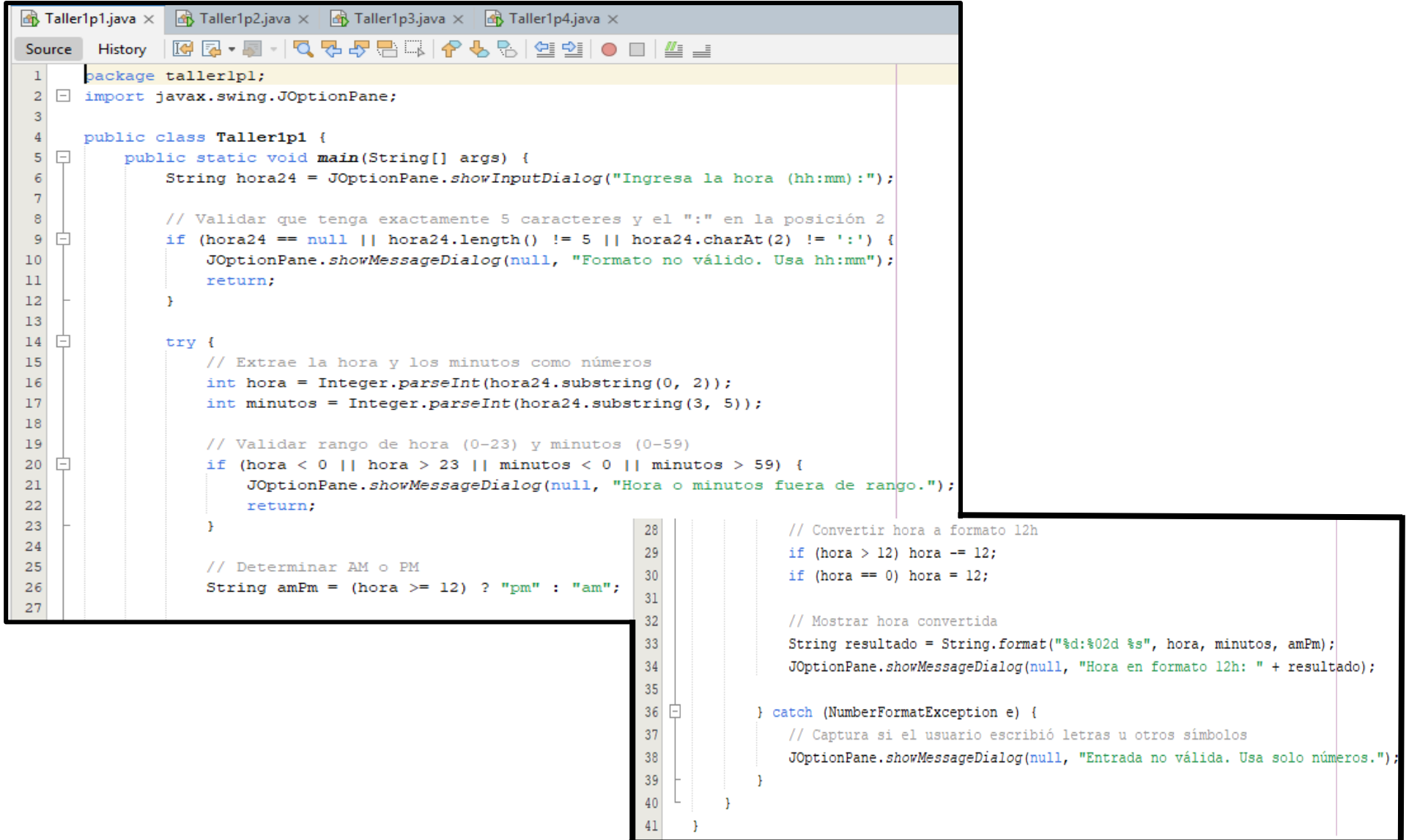
  - a) Contador: con contador para contar hasta 10 (para llevar el registro de cuántos números se han introducido y para determinar cuándo se hayan procesado los 10 números.
  - b) Número: el valor reciente introducido por el usuario.
  - c) Mayor: el número más grande encontrado hasta ahora.

Nota: contemple dentro de su código cálculos con decimales, a su vez 2 cifras significativas después del punto. Utilice JOptionPane (entrada, salida). **Valor 10 puntos**

BUENA SUERTE

## Desarrollo - Creación de los códigos

1. "Escriba un programa que lea la hora en notación de 24 horas y que imprima en notación de 12; por ejemplo, si la entrada es 13:45, la salida será 1:45 pm. El programa debe solicitar al usuario que introduzca exactamente cinco caracteres para especificar una hora."

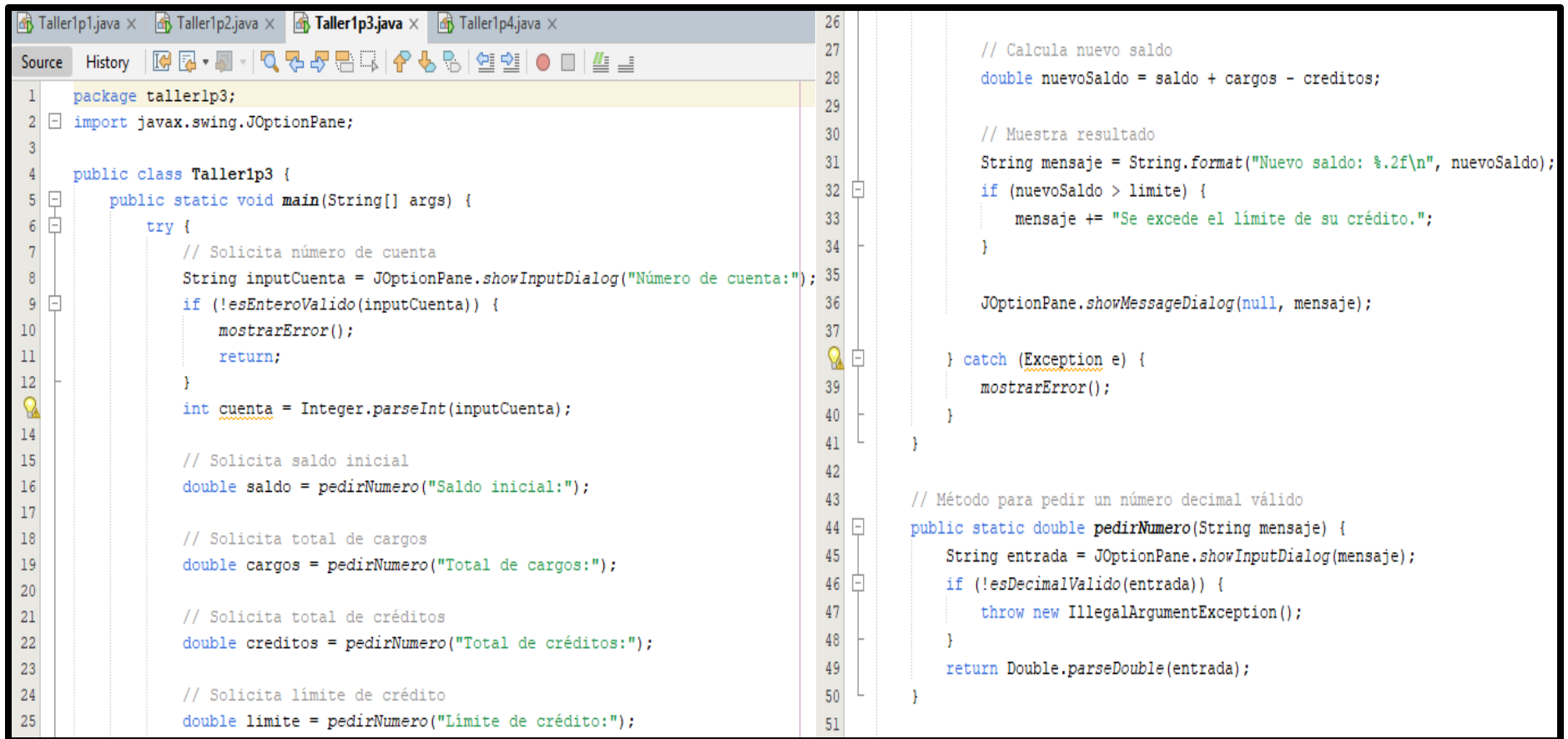


```
1 package taller1p1;
2 import javax.swing.JOptionPane;
3
4 public class Taller1p1 {
5     public static void main(String[] args) {
6         String hora24 = JOptionPane.showInputDialog("Ingresa la hora (hh:mm):");
7
8         // Validar que tenga exactamente 5 caracteres y el ":" en la posición 2
9         if (hora24 == null || hora24.length() != 5 || hora24.charAt(2) != ':') {
10             JOptionPane.showMessageDialog(null, "Formato no válido. Usa hh:mm");
11             return;
12         }
13
14         try {
15             // Extrae la hora y los minutos como números
16             int hora = Integer.parseInt(hora24.substring(0, 2));
17             int minutos = Integer.parseInt(hora24.substring(3, 5));
18
19             // Validar rango de hora (0-23) y minutos (0-59)
20             if (hora < 0 || hora > 23 || minutos < 0 || minutos > 59) {
21                 JOptionPane.showMessageDialog(null, "Hora o minutos fuera de rango.");
22                 return;
23             }
24
25             // Determinar AM o PM
26             String amPm = (hora >= 12) ? "pm" : "am";
27
28             // Convertir hora a formato 12h
29             if (hora > 12) hora -= 12;
30             if (hora == 0) hora = 12;
31
32             // Mostrar hora convertida
33             String resultado = String.format("%d:%02d %s", hora, minutos, amPm);
34             JOptionPane.showMessageDialog(null, "Hora en formato 12h: " + resultado);
35
36         } catch (NumberFormatException e) {
37             // Captura si el usuario escribió letras u otros símbolos
38             JOptionPane.showMessageDialog(null, "Entrada no válida. Usa solo números.");
39         }
40     }
41 }
```

2. "Escribir un programa que lea una cadena de hasta N caracteres que represente un número romano e imprima el formato de ese número y su equivalente en numeración arábica; los caracteres romanos y su equivalente son."

```
Taller1p1.java x Taller1p2.java x Taller1p3.java x Taller1p4.java x
Source History
1 package taller1p2;
2 import javax.swing.JOptionPane;
3
4 public class Taller1p2 {
5     public static void main(String[] args) {
6         String r = JOptionPane.showInputDialog("Número romano (en mayúscula):");
7
8         // Validar que no esté vacío y contenga solo letras romanas válidas
9         if (r == null || r.isEmpty() || !r.matches("[IVXLCDM]+")) {
10             JOptionPane.showMessageDialog(null, "Entrada no válida. Usa solo letras romanas mayúsculas.");
11             return;
12         }
13
14         int total = 0;
15
16         try {
17             for (int i = 0; i < r.length(); i++) {
18                 int actual = valor(r.charAt(i));
19                 int siguiente = (i + 1 < r.length()) ? valor(r.charAt(i + 1)) : 0;
20
21                 if (actual < siguiente) {
22                     total -= actual;
23                 } else {
24                     total += actual;
25                 }
26             }
27
28             JOptionPane.showMessageDialog(null, "Equivalente arábigo: " + total);
29         } catch (Exception e) {
30             JOptionPane.showMessageDialog(null, "Error durante la conversión.");
31         }
32     }
33
34     // Retorna el valor de cada letra romana
35     public static int valor(char c) {
36         switch (c) {
37             case 'I': return 1;
38             case 'V': return 5;
39             case 'X': return 10;
40             case 'L': return 50;
41             case 'C': return 100;
42             case 'D': return 500;
43             case 'M': return 1000;
44             default: throw new IllegalArgumentException("Carácter no válido: " + c);
45         }
46     }
47 }
```

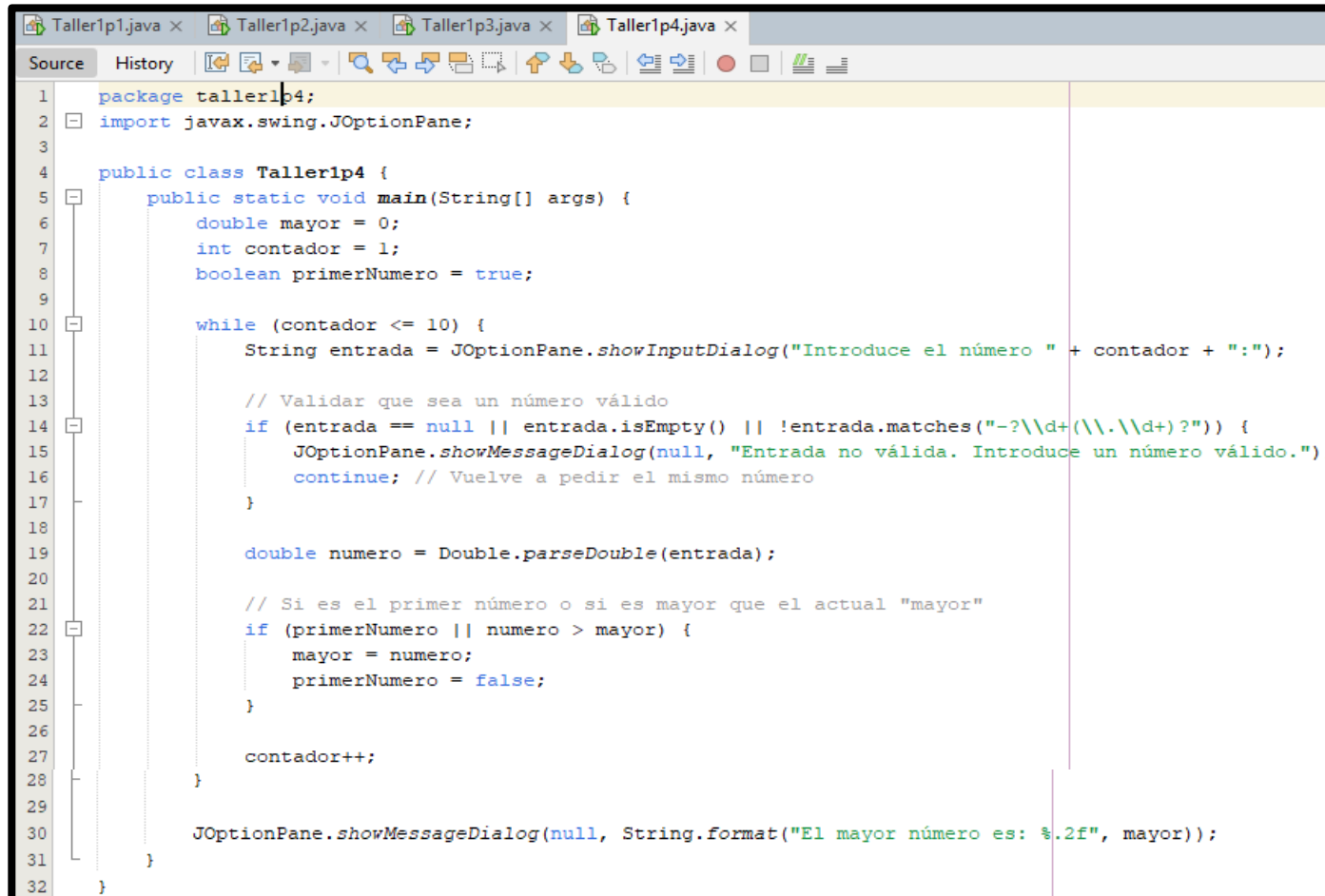
3. “Calculadora de límite de crédito. Desarrolle una aplicación que determine si alguno de los clientes de una tienda de departamentos se ha excedido del límite de crédito en una cuenta.”



```
1 package taller1p3;
2 import javax.swing.JOptionPane;
3
4 public class Taller1p3 {
5     public static void main(String[] args) {
6         try {
7             // Solicita número de cuenta
8             String inputCuenta = JOptionPane.showInputDialog("Número de cuenta:");
9             if (!esEnteroValido(inputCuenta)) {
10                 mostrarError();
11                 return;
12             }
13             int cuenta = Integer.parseInt(inputCuenta);
14
15             // Solicita saldo inicial
16             double saldo = pedirNumero("Saldo inicial:");
17
18             // Solicita total de cargos
19             double cargos = pedirNumero("Total de cargos:");
20
21             // Solicita total de créditos
22             double creditos = pedirNumero("Total de créditos:");
23
24             // Solicita límite de crédito
25             double limite = pedirNumero("Límite de crédito:");
26
27             // Calcula nuevo saldo
28             double nuevoSaldo = saldo + cargos - creditos;
29
30             // Muestra resultado
31             String mensaje = String.format("Nuevo saldo: %.2f\n", nuevoSaldo);
32             if (nuevoSaldo > limite) {
33                 mensaje += "Se excede el límite de su crédito.";
34             }
35
36             JOptionPane.showMessageDialog(null, mensaje);
37
38             } catch (Exception e) {
39                 mostrarError();
40             }
41         }
42
43         // Método para pedir un número decimal válido
44         public static double pedirNumero(String mensaje) {
45             String entrada = JOptionPane.showInputDialog(mensaje);
46             if (!esDecimalValido(entrada)) {
47                 throw new IllegalArgumentException();
48             }
49             return Double.parseDouble(entrada);
50         }
51     }
```

```
52 // Verifica si el valor ingresado es un número entero válido
53 public static boolean esEnteroValido(String texto) {
54     if (texto == null || texto.isEmpty()) return false;
55     return texto.matches("\\d+");
56 }
57
58 // Verifica si el valor ingresado es un número decimal válido
59 public static boolean esDecimalValido(String texto) {
60     if (texto == null || texto.isEmpty()) return false;
61     return texto.matches("\\d+(\\.\\d+)?");
62 }
63
64 // Mensaje de error general
65 public static void mostrarError() {
66     JOptionPane.showMessageDialog(null, "Entrada no válida. Asegúrate de ingresar solo números.");
67 }
68 }
```

4. “Encontrar el número más grande. El proceso de encontrar el valor más grande se utiliza con frecuencia en aplicaciones de computadora.”

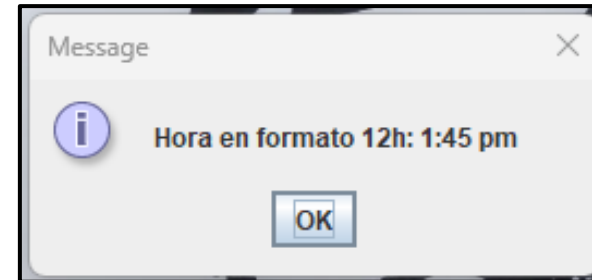
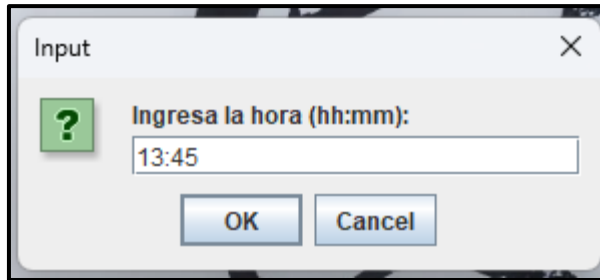


```
1 package taller1p4;
2 import javax.swing.JOptionPane;
3
4 public class Taller1p4 {
5     public static void main(String[] args) {
6         double mayor = 0;
7         int contador = 1;
8         boolean primerNumero = true;
9
10        while (contador <= 10) {
11            String entrada = JOptionPane.showInputDialog("Introduce el número " + contador + ":");
12
13            // Validar que sea un número válido
14            if (entrada == null || entrada.isEmpty() || !entrada.matches("-?\\d+(\\.\\d+)?")) {
15                JOptionPane.showMessageDialog(null, "Entrada no válida. Introduce un número válido.");
16                continue; // Vuelve a pedir el mismo número
17            }
18
19            double numero = Double.parseDouble(entrada);
20
21            // Si es el primer número o si es mayor que el actual "mayor"
22            if (primerNumero || numero > mayor) {
23                mayor = numero;
24                primerNumero = false;
25            }
26
27            contador++;
28        }
29
30        JOptionPane.showMessageDialog(null, String.format("El mayor número es: %.2f", mayor));
31    }
32 }
```

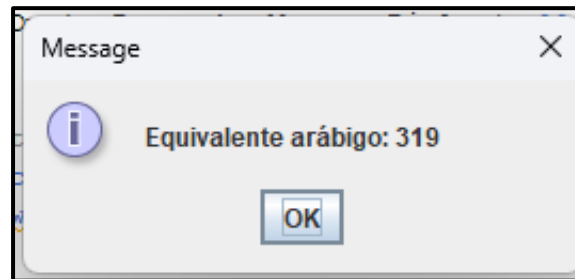
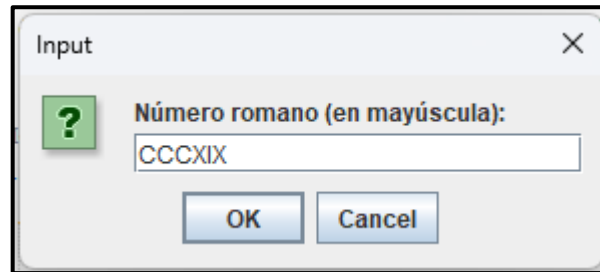
**BONUS:** “contemple dentro de su código cálculos con decimales, a su vez 2 cifras significativas después del punto. Utilice JOptionPane (entrada, salida).”

## Desarrollo – Ejecución de los códigos.

1. Ejecución.



2. Ejecución.



### 3. Ejecución.

The sequence of dialog boxes is as follows:

- Input** (Title: Input):  
? Número de cuenta:  
8  
OK Cancel
- Input** (Title: Input):  
? Saldo inicial:  
300  
OK Cancel
- Input** (Title: Input):  
? Total de cargos:  
300  
OK Cancel
- Input** (Title: Input):  
? Total de créditos:  
200  
OK Cancel
- Message** (Title: Message):  
i Nuevo saldo: 400.00  
Se excede el límite de su crédito.  
OK

### 4. Ejecución.

(Debido a la cantidad de capturas, se escribirán los números aquí: 1, 5, 7, 4, 3, 2, 1, 1, 2, 8)

The dialog box is as follows:

- Message** (Title: Message):  
i El mayor número es: 8.00  
OK