

# Introducción a Ciencias de la Computación

## Tarea 4: Scanner y Controles de flujo

Pedro Ulises Cervantes Gonzáalez  
confundeme@ciencias.unam.mx

Emmanuel Cruz Hernández  
emmanuelcruz@ciencias.unam.mx

Adriana Sánchez Del Moral  
adrisanchez@ciencias.unam.mx

Víctor Zamora Gutiérrez  
agua@ciencias.unam.mx

Fecha de entrega: 13 de abril 2020 11:59pm

1. ¿Para qué sirve los parámetros del main "String arg []"?
2. Escribe 5 métodos de Scanner que nos ayuden a tomar los datos de entrada. Agrega su definición del API de JAVA.
3. ¿En que momento se dispara "java.util.InputMismatchException"?
4. ¿Cuántas veces se itera en los siguientes códigos?  
¿Cual es el valor final de i en cada caso?  
¿Cual es el alcance de la variable i, j en cada caso?  
Si hay error de sintaxis o semántica, ¿Como se corrige?  
Intenta hacerlo sin uso de la programación, lee el código e interprétalo, una vez teniendo tu propio resultado puedes corroborarlo haciendo su ejecución.  
Deja tu resultado y el resultado de la ejecución, explica cual fue tu error.

(a) 

```
int i = 0;
while (i++ < 4){
    System.out.println(i);
}
System.out.println("valor_final_"+i);
```

(b) 

```
int i = 3;
while (i++ < 4){
    System.out.println(i);
}
System.out.println("valor_final_"+i);
```

(c) 

```
for(int i = 0 ; i < 4 ; i++){
    System.out.println(i);
}
System.out.println("valor_final_"+i);
```

(d) 

```
for(int i = 0, j = 0; i < 4 && j < 3; i+=j){
    System.out.println(i);
}
System.out.println("valor_final_"+j);
```

5. En el archivo Calculadora.java adjunto en esta tarea, realiza un menú para el usuario:

NOTA: no modifiques la firma de ningún método ni agregues métodos auxiliares, todo lo siguiente debe ser implementado en el método main.

Tendra un menú que enumere las operaciones que puede realizar, la numeración será la siguiente:

- 0- Salir
- 1- Configura los datos de la calculadora
- 2- Obtener datos
- 3- Hacer suma binaria
- 4- Hacer resta binaria
- 5- Hacer multiplicación
- 6- Hacer division
- 7- Hacer modulo
- 8- Hacer suma operador incremento postfixo
- 9- Hacer suma operador incremento sufijo
- 10- Hacer resta operador incremento postfixo
- 11- Hacer resta operador incremento sufijo

Este menú se imprimirá un numero indeterminado de veces hasta que la entrada de la opción seleccionada del usuario sea cero, esto **romperá** el ciclo y terminara la ejecución. Analiza que tipo de control de flujo es mejor para esta implementación.

Descripcion de las opciones del menú:

La opción seleccionada deberá ser evaluada dentro de la estructura de control de flujo "Switch", si utilizas alguna otra, describe porque tu selección es mejor que un switch para este ejercicio.

- (a) Salir  
Termina la ejecución del ciclo y por lo tanto la ejecución del código
- (b) Configura los datos de la calculadora  
Modifica los valores color, marca y dueño atributos de la calculadora con los nuevos datos ingresados por el usuario.
- (c) Obtener datos  
Muestra la información de la calculadora, utiliza el método *toString()* ya implementado en el codigo.
- (d) Opciones 3 a 11  
Realiza la operación seleccionada utilizando dos valores ingresados por el usuario. Imprime el resultado con el siguiente formato

*"Resultado - > " + resultado*

Recuerda que en la firma de los métodos los parámetros utilizados son de tipo entero.

Utiliza la instancia **calculadora** para realizar la ejecución de tus métodos.

Una vez que el usuario haya seleccionado la opción cero, la calculadora debe imprimir el siguiente mensaje de agradecimiento:

*"Calculadora de" + nombredeldueño + " fuera."*

Esta impresión debe hacerse fuera de su estructura de control que hayan utilizado

Responde lo siguiente

- (a) ¿Por qué utilizaste esa estructura de control para el manejo del menú?
- (b) Si utilizaste una estructura diferente al switch, ¿Por qué esa implementación es mejor?
- (c) ¿Cómo puedes mejorar la robustez de tu código o que medidas tomaste para robustecerlo?  
Aclaro, no estoy pidiendo la implementación de excepciones en java.