

Capítulo III – Escribiendo simples programas

Trabajaremos con proyectos Maven

Parte I: Entradas de consola

Interactivo: acepta entradas en la consola

No es mucho, pero es un pequeño paso

```
Integer miEntradaEntero= Integer.parseInt(args[0]);
Double miEntradaDecimal = Double.parseDouble(args[0]);
Boolean miVariableBool=Boolean.parseBoolean(args[0]);
char miVariableCaracter = args[0].charAt(0);
```

```
for (int i = 0; i < args.length; i ++) {
    System.out.println("El entero " + args[i]);
    Integer miEntrada =
Integer.parseInt(args[i]);
    Chequeo.ChequeoPar(miEntrada);
    System.out.println();
}
```

Podemos descifrar arreglos

Un programador excéntrico y ligeramente loco dijo que todo profesional usa el terminal de comando, entonces aprovechémonos de esta oportunidad

```
int[] miArreglo = new int[args.length];
    for (int i = 0; i < args.length; i ++) {
        Integer miEntrada =
Integer.parseInt(args[i]);
        miArreglo[i] = miEntrada;
    }
    int miSuma = Chequeo.SumarArreglo(miArreglo);
    System.out.println(miSuma);
}
```

¿Cómo editarías la función para que te calcule el promedio de tus notas?

```
javac App.java
```

```
java App 10 11 11
```

Resumen:


- Con entradas de consola, podemos decir que esto se parece más a un programa ya que es más interactivo
- Cuidado con los tipos de variables hay que descifrarlo a tal punto que esté empatado con el tipo de mi función
- Hemos visto cómo bucles, tipos de variables,

función chequeo par

chequeo impar

Parte II: Entradas de usuario

orientativo

Fórmula	No incluye
$v_f = v_0 \pm gt$	Sin h
$h = v_0 t \pm \frac{gt^2}{2}$	Sin v_f
$h = \left(\frac{v_0 + v_f}{2}\right) \cdot t$	Sin g
$v_f^2 = v_0^2 \pm 2gh$	Sin t
$h_n = v_0 \pm \frac{g}{2}(2n - 1)$	

<https://matemovil.com/wp-content/uploads/2020/01/f%C3%B3rmulas-ca%C3%ADda-libre.jpg>

```
// 1. inicializar escáner y tomar entradas
Scanner entradaUsuario = new
Scanner(System.in);

System.out.println("Ingresa el mes: ");
String mes = entradaUsuario.next();

System.out.println("Ingresa el día: ");
int dia = entradaUsuario.nextInt();
```

```
        // 2. código
        System.out.println("Hoy es " + dia + " de " +
mes );

        // 3. cerrar el scanner ~= colgando telefono
        entradaUsuario.close();
```

```
Scanner entradaUsuario = new Scanner(System.in);
    System.out.println("Hola, Persona cruel, ingresa lo
que te falta: ");
    //char entrada = entradaUsuario.next().charAt(0);
    while (entradaUsuario.hasNext()){
        //char entrada =
entradaUsuario.next().charAt(0);
        String entrada = entradaUsuario.next();
        //char entradaProcesada =
Character.toUpperCase(entrada);
        //System.out.println(entrada);
        if ( entrada.equalsIgnoreCase("altura")) {
            Scanner entradaUsuarioAnidado = new
Scanner(System.in);
            System.out.println("Ingresa tu velocidad
inicial y el tiempo de caída (separado por un espacio):
");
            //
            System.out.println(entradaUsuarioAnidado.hasNext());
            double miV0 =
entradaUsuarioAnidado.nextDouble();
            double miTiempo =
entradaUsuarioAnidado.nextDouble();
            double miVelocidadFinal = VFSinH(miV0,
miTiempo);

            // entradaUsuario.close();
            System.out.println("La velocidad final de
impacto es de: " + miVelocidadFinal + " m/s");
```

```

        //entradaUsuarioAnidado.close();
    }

    else if ( entrada.equalsIgnoreCase("tiempo")) {
        Scanner entradaUsuarioAnidado = new
Scanner(System.in);
        System.out.println("Ingresa tu velocidad
inicial y la altura de la caída (separado por un
espacio): ");
        double miV0 =
entradaUsuarioAnidado.nextDouble();
        double miAltura =
entradaUsuarioAnidado.nextDouble();
        double miVelocidadFinal = VFSinT(miV0,
miAltura);

        // entradaUsuario.close();
        System.out.println("La velocidad final de
impacto es de: " + miVelocidadFinal + " m/s");
        //entradaUsuarioAnidado.close();

    }

    else if (entrada.equalsIgnoreCase("para")) {
        System.out.println("El programa terminó");
        break;
    }

}

entradaUsuario.close();

}

```

Puedes expandir este pequeño programa

Si la gravedad no es 9.81 ms⁻¹

Si la caída es 42 metros/segundo

Diferentes unidades (km/min)

Intenta con tu propio programa de la situación de la vida real: lista de compras calcular el precio de descuento, nota necesario para, coleccionar datos, entrevista, entre otros.

Enfatizar que esta no es la única esquematización:

Pudiste definir una función VelFinal y tomar un argumento

Pudiste obtener todas las entradas primero

Hay bastante libertad sobre cómo diseñar el programa

Intenta en Python, usando la función eval(input())

Resumen

- Usar la clase SCANNER para leer entradas personalizadas del usuario (practicar orientación)
- Hemos visto bucles mientras, estructuras de control

Escribir código mas robusto

Capítulo III.III – Escribiendo código robusto

¿Qué es un programa?

Esta puede recibir varias definiciones, sin embargo, el que nosotros podemos seguir es que un programa es implementado a través de funciones que usualmente requiere en vivo entradas del usuario.

Antes, trabajábamos con código que se veía de esta manera:

```
if __name__ == "__main__": # similar a public static void main
...
    es_impar(12)
    es_impar(2)
    es_impar(-1)
```

Egreso:

(base) C:\Users\86136\Desktop\club\proyectosPython>python funcion.py

es par

es par

es dos, un primo

es impar

No es muy interactivo. Sin embargo, podemos modificar esta función con el fin de que el usuario pueda usar la entrada que quiera.

```

if __name__ == "__main__": # similar a public static void main
...
    print("Bienvenidos a este programa que chequea si tu entero
o es impar o no :)")
    n = eval(input("Ingresa el entero: "))
    es_impar(n)

```

Egreso:

(base) C:\Users\86136\Desktop\club\proyectosPython>python funcion.py

Bienvenidos a este programa que chequea si tu entero es impar o no :)

Ingresa el entero: -1

es impar

Lo que estamos haciendo ahora es *evaluar* las entradas del usuario.

Debido que la naturaleza de los programas es ser orientativo, si escribes un programa es probable que uses la función *print* de manera consecutiva y más frecuente que *return*. Puedes por ejemplo usar *.format* en Python si quieres imprimir algo que contiene una variable.

```

Scanner entradaUsuario = new Scanner(System.in);
    System.out.println("Hola, Persona cruel, ingresa lo
que te falta: ");
    //char entrada = entradaUsuario.next().charAt(0);
    while (entradaUsuario.hasNext()){
        //char entrada =
entradaUsuario.next().charAt(0);
        String entrada = entradaUsuario.next();
        //char entradaProcesada =
Character.toUpperCase(entrada);
        //System.out.println(entrada);
        if ( entrada.equalsIgnoreCase("altura")) {
            // Scanner entradaUsuarioAnidado = new
Scanner(System.in);
            try {
                System.out.println("Ingresa tu
velocidad inicial y el tiempo de caída (separado por un
espacio): ");
                //
                System.out.println(entradaUsuarioAnidado.hasNext());

```

```

        double miV0 =
entradaUsuario.nextDouble();
        double miTiempo =
entradaUsuario.nextDouble();
        if (miTiempo < 0 || miV0 < 0) {
            throw new
IllegalArgumentException("Esto es caída libre, no
puedes ingresar velocidad o tiempo negativo.");
        }
        double miVelocidadFinal = VFSinH(miV0,
miTiempo);

        // entradaUsuario.close();
        System.out.println("La velocidad final
de impacto es de: " + miVelocidadFinal + " m/s");
    } catch (IllegalArgumentException error) {
        System.out.println("Ingresa un nuevo
valor positivo");
        double miV0 =
entradaUsuario.nextDouble();
        double miTiempo =
entradaUsuario.nextDouble();

        double miVelocidadFinal = VFSinH(miV0,
miTiempo);

        // entradaUsuario.close();
        System.out.println("La velocidad final
de impacto es de: " + miVelocidadFinal + " m/s");
    }

    //entradaUsuarioAnidado.close();
}

else if ( entrada.equalsIgnoreCase("tiempo")) {
    //Scanner entradaUsuarioAnidado = new
Scanner(System.in);

```

```

        System.out.println("Ingresa tu velocidad
inicial y la altura de la caída (separado por un
espacio): ");
        double miV0 = entradaUsuario.nextDouble();
        double miAltura =
entradaUsuario.nextDouble();
        double miVelocidadFinal = VFSinT(miV0,
miAltura);

        // entradaUsuario.close();
        System.out.println("La velocidad final de
impacto es de: " + miVelocidadFinal + " m/s");
        //entradaUsuarioAnidado.close();

    }
    else if (entrada.equalsIgnoreCase("para")) {
        System.out.println("El programa terminó");
        break;
    }
}

```

Intenta completar el código con try-catch para la otra condición.

```

Scanner entradaUsuario = new Scanner(System.in); //
cogemos el telefono
        System.out.println("Ingresa 2 decimales que van
a ser divididos (separados por un espacio)");

        while (entradaUsuario.hasNext()){

            try {
                // System.out.println("Ingresa 2
decimales que van a ser divididos (separados por un
espacio)");

                double miNumerador =
entradaUsuario.nextDouble();
                double miDenominador =
entradaUsuario.nextDouble();

```



```

        double resultado =
miNumerador/miDenominador;

        System.out.println("Excelente! El
resultado es: " + resultado);

    } catch (InputMismatchException error) {
        System.err.println("Es probable que no
ingresaste un decimal.");
        main(args);
    }
}

entradaUsuario.close(); // colgamos el telefono

```

Inténtalo en Python

Resumen:

- Esquematizar antes de programar con el fin de anticipar errores
- Código robusto también se trata de escribir un programa que no crashee y tolere argumentos ilegales.

Claro que, eres libre de decidir si quieres escribir funciones más interactivas, esas que requieren entradas de usuario. En realidad, como hemos visto, podemos ingresar la entrada previo a la ejecución de código.

Capítulo 2.1: Consejos de programación

Hasta este punto he tratado de introducirte diferentes conceptos de programación, sean estas estructuras de control o bucles. Hoy vamos a explorar diferentes consejos con el fin de emplear estos conceptos de manera eficiente.

Planeación antes de ejecución

Tal como escribir un ensayo, escribir código requiere un proceso de planteamiento previo a la ejecución de código. Una estrategia común es escribir pseudo código, algo que respeta en cierto punto la sintaxis de un lenguaje de programación, sin embargo al mismo tiempo no es código en sí. En el previo capítulo ya hemos visto algo similar:

```

for (int i = 0; i < args.length; i++) {
    try {
        System.out.println("El entero " +
args[i]);
    }
}

```

```
        Integer miEntrada =  
Integer.parseInt(args[i]);  
        Chequeo.ChequeoPar(miEntrada);  
        System.out.println();} catch  
(NumberFormatException error) {  
            System.out.println("Por favor  
ingresa un entero la proxima vez!");  
        }  
    }
```