Práctica Guiada (Ejercicios)

# OBJETIVO

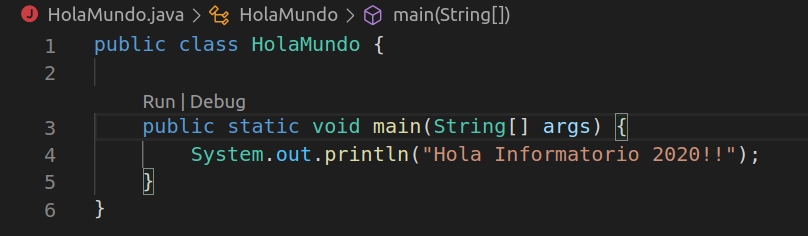
El objetivo de este apunte es dar una guía para la resolución de problemas y sintaxis de Java. Para hacer una revisión rápida, los ejercicios no realizarán validaciones de los datos de entrada (ejemplo, si nos pide que ingresemos por pantalla un numero y le ingresamos una letra) que más adelante en el curso lo revisaremos.

# EJERCICIOS

1. Realizar un programa que realice la impresión por consola del siguiente mensaje:

|  |
| --- |
| ***Hola Informatorio 2020!!*** |

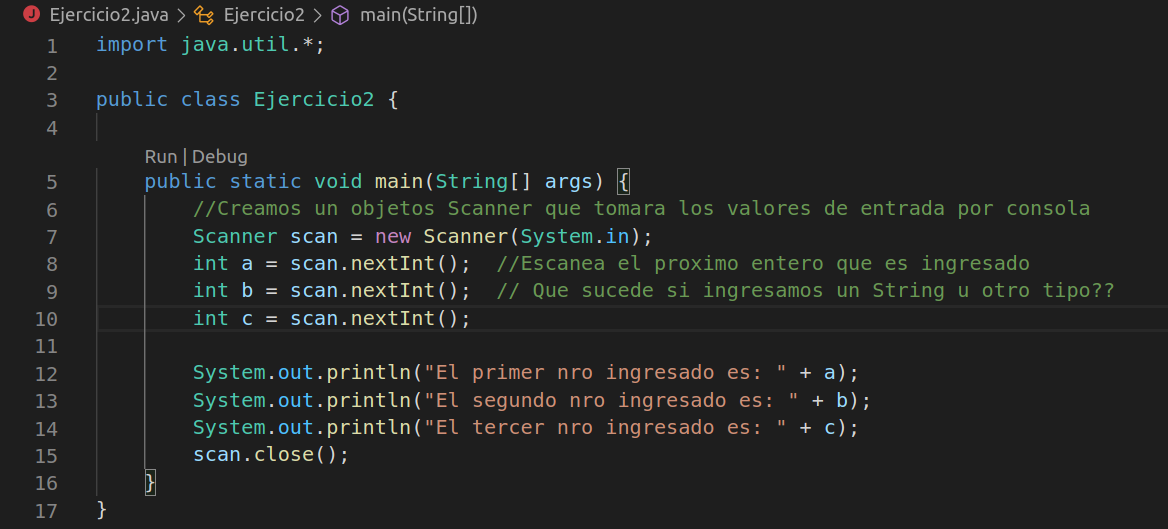
**Solución**:



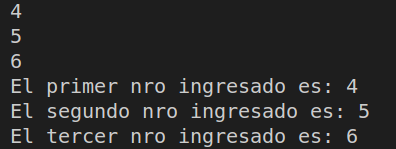
1. Realizar un programa que al ejecutar se pueda ingresar 3 numeros enteros, para luego imprimirlos por pantalla de la siguiente manera:

|  |
| --- |
| **El primer nro ingresado es: 4**  **El segundo nro ingresado es: 5**  **El tercer nro ingresado es: 6** |

**Solución**:



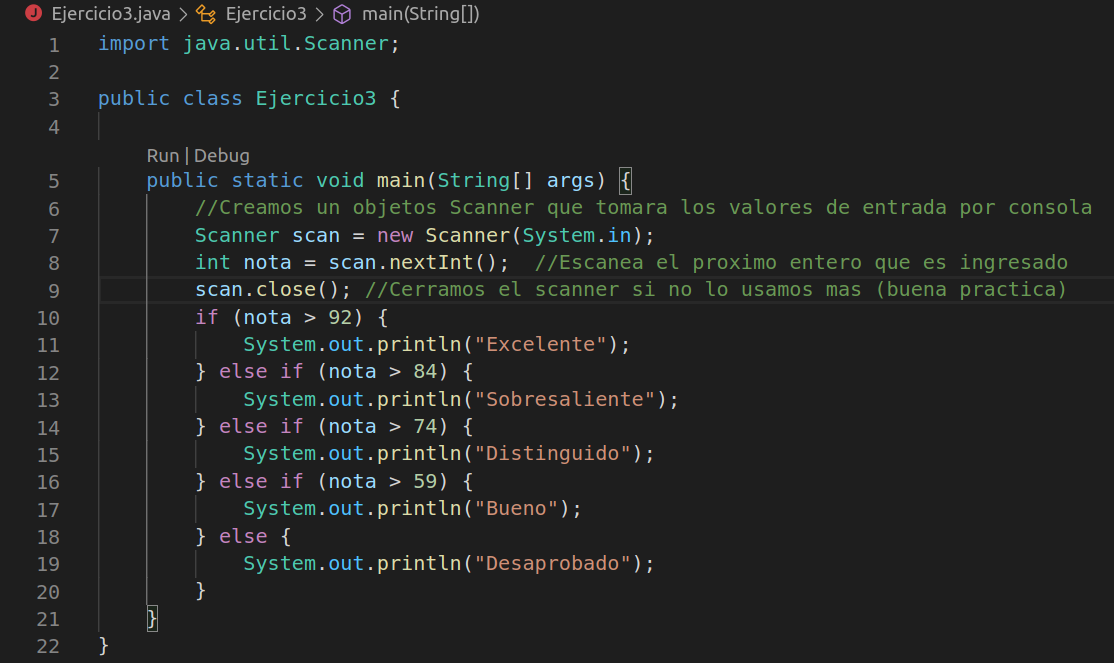
**Test por consola**:



1. Tendremos un sistema que debe clasificar el resultado de una evaluación. Como dato de entrada tendremos un entero y como resultado un String de la clasificación de la siguiente forma:

|  |  |
| --- | --- |
| **Rango** | **Clasificación** |
| 93 - 100 | Excelente |
| 85 - 92 | Sobresaliente |
| 75 - 84 | Distinguido |
| 60 - 74 | Bueno |
| 00 - 59 | Desaprobado |

**Solución (if-else)**:



|  |
| --- |
| Podemos ver el uso de la sintaxis condicional **if-else** como lo vimos en Python. Pueden existir otras soluciones aceptables y que resuelvan el problema también (por ejemplo, en vez de preguntar por mayor podemos hacerlo por menor).También vemos el uso de operadores lógicos. Debajo se adjunta otros operadores de comparación y lógicos comunes para que puedas usarlos en otros ejercicios propuestos |

|  |  |
| --- | --- |
| **Operador** | **Significado** |
| == | Igualdad |
| != | Distinto (No igual) |
| > | Mayor |
| >= | Mayor o Igual |
| < | Menor |
| <= | Menor o Igual |

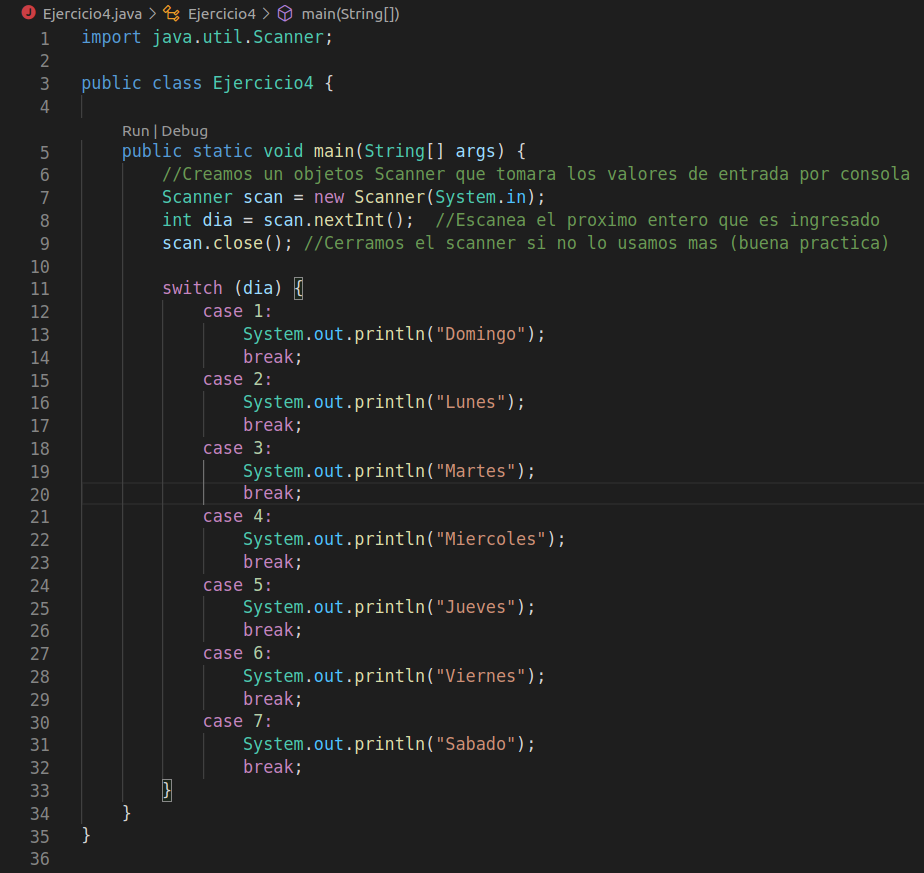
1. Realizar un programa que dado un número de entrada entre 1 a 7, nos devuelva el dia de la semana de la siguiente forma:

|  |  |
| --- | --- |
| **Número** | **Dia** |
| 1 | Domingo |
| 2 | Lunes |
| 3 | Martes |
| 4 | Miercoles |
| 5 | Jueves |
| 6 | Viernes |
| 7 | Sabado |

**Solución 1 (if-else)**:

Podemos desarrollar un programa parecido al ejercicio anterior número 3. Intentalo!!

**Solución 2 (switch)**:



|  |
| --- |
| En este caso, podremos observar la sentencia ***switch*** (también llamada ***según***). La funcionalidad que expone esta sentencia se basa según un caso, realiza una operación.  Prestemos atención a la sentencia ***break***.  ¿Qué sucedería si eliminamos todas esas sentencias ***break***?  ¿Que imprimirá si luego ejecuto el programa e ingreso el número 4?. |

1. Desarrollar un programa que dado un número entero ingresado, nos retorne los valores de ese número multiplicado por 1, 2, 3, … y 10.

Ejemplo para el dato de entrada: 2

Salida:

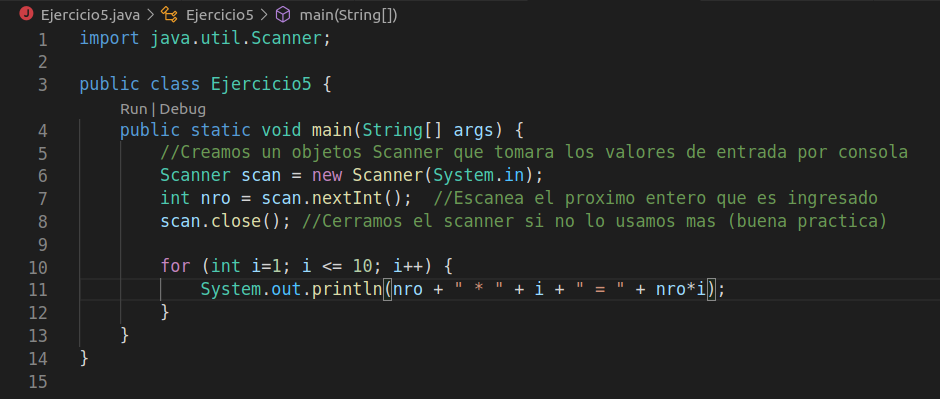
|  |
| --- |
| **2 \* 1 = 2**  **2 \* 2 = 4**  **2 \* 3 = 6**  **2 \* 4 = 8**  **2 \* 5 = 10**  **2 \* 6 = 12**  **2 \* 7 = 14**  **2 \* 8 = 16**  **2 \* 9 = 18**  **2 \* 10 = 20** |

**Solución (*for*)**:

Aplicaremos una de las estructuras condicionales que usamos también en el módulo de Python que es el bucle ***for*** (o también llamado ***Para***).

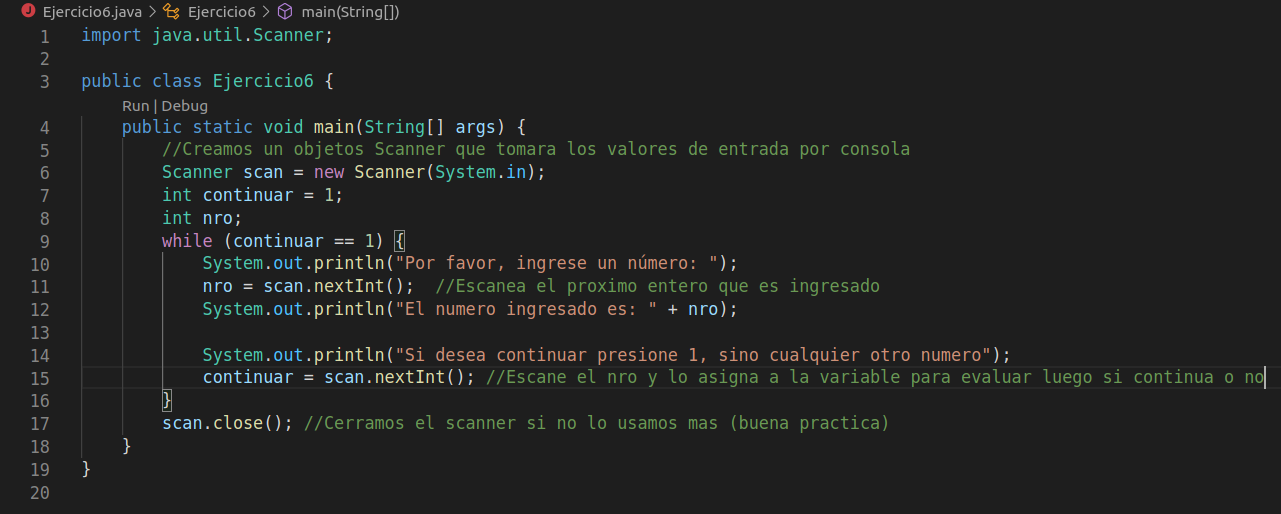
La sintaxis de for para uso general es de la siguiente manera:

**for (inicialización; terminación; incremento)**

****

1. Debemos realizar un programa que nos realizará una pregunta si queremos realizar una operación dada o si deseamos salir/terminar el programa.

Utilizaremos en este caso una simple operación que nos pide un número y luego lo imprime.

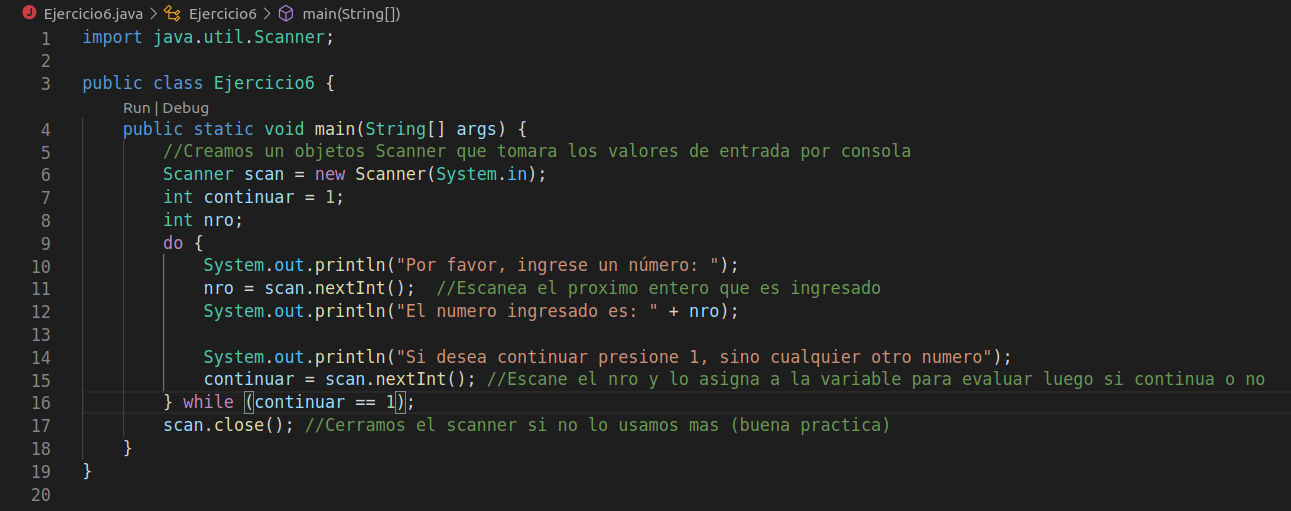


|  |
| --- |
| **Ahora debes hacerlo tu**:  Siguiendo este ejemplo, trata de mejorar el código de los ejercicios 2, 3, 4 y 5. Donde realizaremos una vez la operación y luego preguntaremos si continuar con otro cálculo. Caso contrario, finaliza el programa. |

**Otra variante** usando la sentencia ***do-while***:

En la sentencia ***while***, l**a condición** para ejecutar un ciclo del bucle **se evalúa al principio**. De esta forma, es posible que dicha evaluación sea falsa desde el principio y el bucle itera 0 veces (o no itere).

En cambio con la sentencia ***do-while* evalúa la condición de fin al final del bucle**, quiere decir que **el bucle mínimamente iterara 1 vez**. En el siguiente ejemplo se realiza la misma versión del código anterior pero con el bucle do-while.



1. En base a los ejercicios 2 al 4, modificar los programas para que separe la lógica de negocio (cálculo aritmético, procesamiento, etc) en funciones.  
     
   Ejemplo:  
   