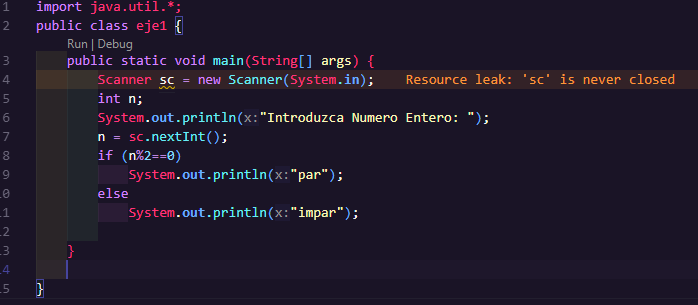
**Cuadernillo de prácticas java NetBeans**

**Informe de Ejercicios Básicos con Estructura Condicional**

**SENA**

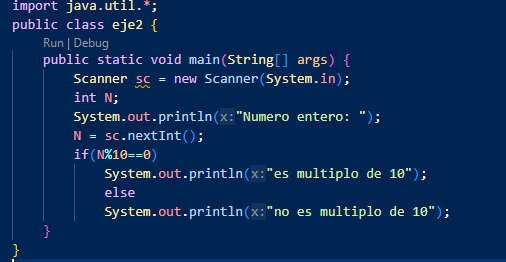
**Bogotá D.C**

**Ejercicio 1:** Programa Java que lea un número entero por teclado y calcule si es par o impar. Podemos saber si un número es par si el resto de dividir el número entre 2 es igual a cero. En caso contrario el número es impar.



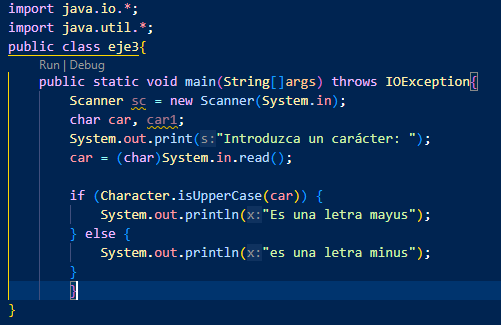
* Import java.util.\*; esto importa las clases del paquete java.util
* Public static es la entrada del programa el que lo ejecuta
* Scanner sc = new Scanner(System.in); crea el objeto sc tipo Scanner(captura datos del tipo int, double, string y etc.) que permite leer la entrada de datos que registra el usuario, System.in indica que los datos se leerán desde la entrada estándar (el teclado).
* n = sc.nextInt(); se le asigna a n el numero entero que digita el usuario, nextInt() lee el siguiente número de entrada.
* if (n%2==0) utiliza el condicional if para saber si el numero es par o impar, se utiliza el operador mod(%) que obtiene el residuo de la división de n.

**Ejercicio 2:** Programa que lea un número entero y muestre si el número es múltiplo de 10.



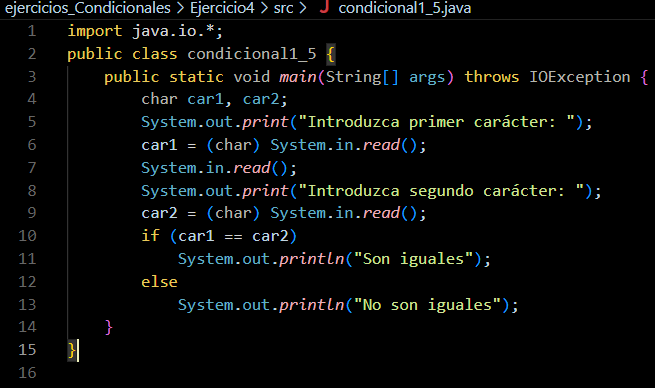
* Scanner sc = new Scanner(System.in); **Crea un objeto Scanner para entrada**: La clase Scanner se utiliza para capturar la entrada del usuario. El parámetro System.in indica que se capturará desde la consola.
* System.out.println("Numero entero: "); **Muestra un mensaje en la consola**: Imprime el texto "Numero entero: " para que el usuario sepa que debe ingresar un número.
* N = sc.nextInt(); **Captura la entrada del usuario**: Utiliza el método nextInt() del objeto Scanner para leer un número entero ingresado por el usuario y lo almacena en la variable N.
* if (N % 10 == 0) verifica si el número ingresado (N) es múltiplo de 10. Esto se hace calculando el residuo de N entre 10 (N % 10), y si el resultado es 0, significa que N es un múltiplo de 10.

**Ejercicio 3:** Programa que lea un carácter por teclado y compruebe si es una letra mayúscula.

****

* **Importa las librerías necesarias**: java.io.\* y java.util.\* son bibliotecas de Java. La primera incluye clases para la entrada/salida como IOException y métodos de lectura como System.in.read(). La segunda incluye utilidades como Scanner.
* public static void main(String[] args) throws IOException { **Método principal main con excepción**: Este es el punto de entrada del programa. El throws IOException indica que el método puede lanzar una excepción si hay un error de entrada/salida, específicamente relacionado con la lectura de datos.
* char car, car1; **Declara dos variables de tipo char**: Se declaran las variables car y car1 de tipo carácter.
* car = (char)System.in.read(); **Lee el carácter ingresado por el usuario**: Utiliza System.in.read() para leer el valor del carácter. Como System.in.read() devuelve un entero, se hace un casting explícito a char para convertirlo a un carácter.
* if (Character.isUpperCase(car)) { **Condicional para verificar si el carácter es mayúscula**: Usa el método estático isUpperCase de la clase Character para verificar si el carácter ingresado es una letra mayúscula. Si lo es, ejecuta el bloque if.

**Ejercicio 4**: Programa que lea dos caracteres y compruebe si son iguales.

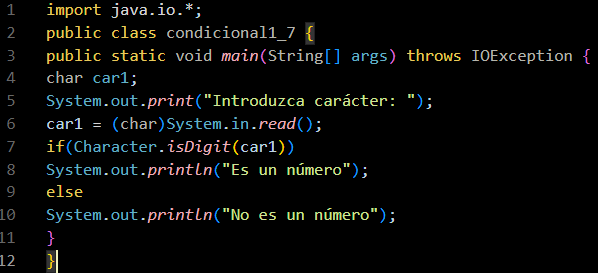


* **Imports**: Se importa la clase java.io.\* para poder utilizar la funcionalidad de entrada y salida de datos, necesaria para leer los caracteres ingresados.
* El programa se ejecuta dentro de la clase **condicional1\_5**, y la función **main** es el punto de entrada del programa.
* Se declara que el método **main** puede lanzar una excepción de entrada/salida **(IOException)** debido al uso de la función **System.in.read().**
* Se declaran dos variables de tipo **char, car1 y car2,** para almacenar los caracteres que el usuario ingresará.
* El programa solicita al usuario que ingrese el primer carácter, lo lee usando **System.in.read(),** y lo guarda en **car1.**
* Luego se llama a **System.in.read()** de nuevo para descartar el carácter de "intro" (Enter) que se queda en el búfer después de la primera lectura.
* El programa repite el proceso para el segundo carácter, guardándolo en car2.

**Condición:**

El programa compara si **car1** es igual a **car2** usando el operador ==. Si son iguales, imprime "Son iguales", y si no, imprime "No son iguales".

**Ejercicio 5**: Una forma alternativa de hacer este programa es creando dos objetos Character a partir de los caracteres que se han leído y compararlos utilizando el método equals.

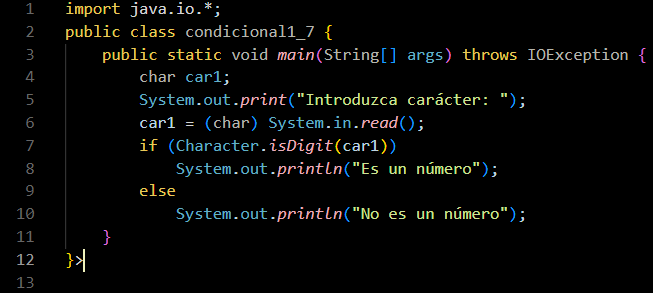


Este código Java es una versión modificada del anterior, que introduce el uso de la clase Character para comparar los caracteres en lugar de hacerlo directamente con el operador ==. Aquí están los cambios y su descripción

**Diferencias:**

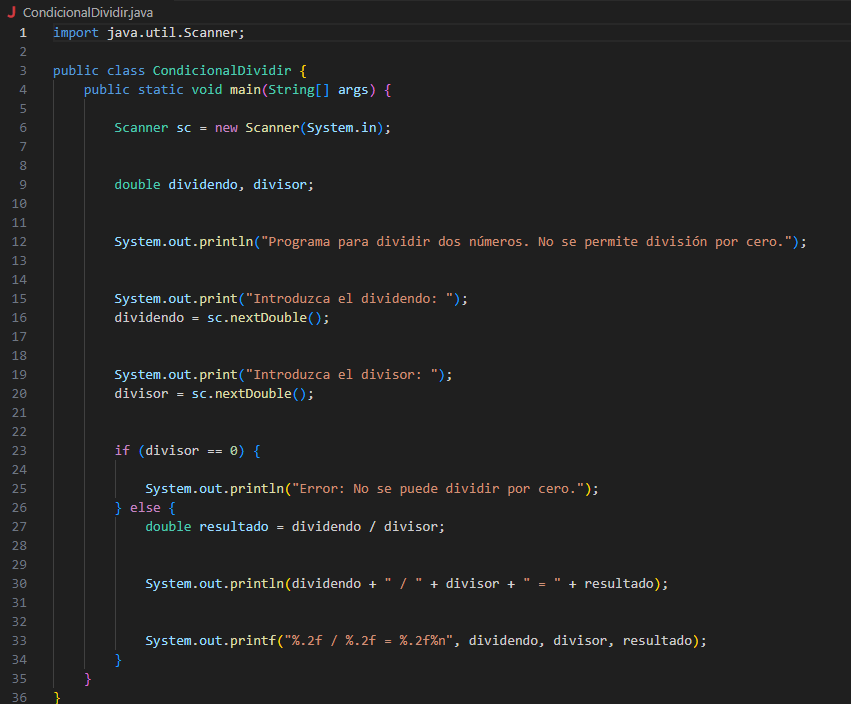
* **Uso de Character en lugar de char:** Ahora los caracteres se envuelven en objetos de la clase Character, lo que permite utilizar el método equals() para realizar la comparación.
* **Método equals():** Se utiliza el método equals() en lugar de == para comparar los dos objetos Character, que es la forma correcta de comparar objetos en Java.

**Ejercicio 6:** Programa que lea dos números por teclado y muestre el resultado de la división del primer número por el segundo. Se debe comprobar que el divisor no puede ser cero.



* Se importa java.io.\* para permitir la entrada de datos desde el teclado.
* El método principal main puede lanzar una excepción **IOException** debido al uso de **System.in.read().**
* Se declara una variable de tipo **char** llamada car1 para almacenar el carácter ingresado por el usuario.
* El programa muestra un mensaje pidiendo al usuario que introduzca un carácter. Después de que el usuario lo hace, el carácter se lee y se guarda en car1 utilizando **System.in.read().**
* El programa verifica si el carácter ingresado es un número utilizando el método estático **Character.isDigit(car1**). Este método devuelve true si el carácter es un dígito (del 0 al 9).
* Si la condición es verdadera, imprime "Es un número".
* Si la condición es falsa, imprime "No es un número"

**Ejercicio 7:** Programa que lea dos números por teclado y muestre el resultado de la división del primer número por el segundo. Se debe comprobar que el divisor no puede ser cero.



* **Importación de la clase Scanner:** El programa inicia importando la clase Scanner desde la librería estándar de Java. Esto permite al programa leer datos introducidos por el usuario a través de la consola.
* **Definición de la clase y el método principal:** A continuación, se define una clase pública llamada CondicionalDividir. Dentro de esta clase, se encuentra el método main, que es el punto de entrada del programa. Este método es donde se ejecutan todas las instrucciones del código.
* **Creación del objeto Scanner:** Se crea un objeto de tipo Scanner, que será utilizado para capturar los valores que el usuario introduce a través de la consola. Este objeto está vinculado a la entrada estándar del sistema.
* **Declaración de variables:** El programa declara dos variables de tipo double, dividendo y divisor, que se utilizarán para almacenar los números que el usuario ingresa para la operación de división. El tipo double se utiliza porque permite manejar números decimales.
* **Mostrar mensaje inicial:** Antes de solicitar los valores, el programa informa al usuario sobre su propósito mediante un mensaje que indica que se va a realizar una división y que no es posible dividir entre cero. Esto proporciona contexto para que el usuario entienda qué debe hacer.
* **Solicitud de los valores:** El programa solicita al usuario que introduzca el valor del dividendo, que es el número que se va a dividir, y luego le pide el valor del divisor, que es el número por el cual se va a dividir. Ambos valores se capturan mediante el objeto Scanner.
* **Validación del divisor:** Una vez obtenidos los valores, el programa verifica si el divisor es igual a 0. Si el divisor es 0, se muestra un mensaje de error, ya que la división por cero no es permitida ni en matemáticas ni en programación, lo que podría generar un error en tiempo de ejecución.
* **Cálculo y muestra del resultado:** Si el divisor no es cero, el programa realiza la operación de división y guarda el resultado en una variable. Luego, muestra este resultado de dos maneras: primero con un mensaje simple, y luego utilizando un formato que limita el número de decimales a dos para una mayor precisión en la presentación del resultado.

Texto

Descripción generada automáticamente**Ejercicio 8:** El programa lee por teclado tres números enteros y calcula y muestra el mayor de los tres.

* **Importación de la clase Scanner:** El programa comienza importando la clase Scanner de la librería java.util. Esto es esencial para permitir que el programa lea entradas desde el teclado. La clase Scanner es parte de la API estándar de Java y facilita la captura de datos introducidos por el usuario.
* **Definición de la clase y método principal:** El programa define una clase pública llamada MayorDeTres. Esta clase contiene el método main, que es el punto de entrada del programa. Es donde se ejecutan todas las instrucciones para encontrar el mayor de tres números ingresados por el usuario.
* **Creación del objeto Scanner:** Se crea un objeto de tipo Scanner con la línea Scanner sc = new Scanner(System.in);. Este objeto permite leer los valores ingresados por el usuario desde la consola. Aquí se especifica que se leerán datos desde la entrada estándar (System.in).
* **Declaración de las variables n1, n2, y n3:** El programa declara tres variables enteras (int) llamadas n1, n2, y n3. Estas variables almacenarán los tres números ingresados por el usuario, que se compararán entre sí para determinar cuál es el mayor.

**Solicitud de los números:**

* El programa solicita al usuario que introduzca el primer número con System.out.print("Introduzca primer número: "); y luego almacena el valor en la variable n1 con n1 = sc.nextInt();.
* Después, se repite el proceso para el segundo y el tercer número, almacenando los valores en n2 y n3, respectivamente.

**Comparación de los números:** El corazón del programa radica en una estructura if-else anidada que compara los tres números introducidos:

* **Primera comparación (n1 > n2)**: Si el primer número es mayor que el segundo, se procede a comparar n1 con n3 para verificar si también es mayor que el tercer número:
  + Si n1 > n3, entonces el primer número es el mayor y se imprime el mensaje correspondiente.
  + Si n1 <= n3, entonces el tercer número es el mayor y se imprime como tal.
* **Segunda comparación (n2 > n3)**: Si el primer número no es mayor que el segundo, se verifica si el segundo número es mayor que el tercero:
  + Si n2 > n3, se imprime que el segundo número es el mayor.
  + De lo contrario, se imprime que el tercer número es el mayor.
* **Salida del resultado:** Dependiendo de las comparaciones realizadas en la estructura condicional, el programa imprimirá el mayor de los tres números en la consola usando System.out.println(). El mensaje indicará cuál de los tres números ingresados es el mayor, junto con su valor.

**Ejercicio 9:** Programa que lea por teclado tres números enteros H, M, S correspondientes a hora, minutos y segundosrespectivamente, y comprueba si la hora que indican es una hora válida.

Texto

Descripción generada automáticamente

* **Importación de la clase Scanner:** El programa comienza importando la clase Scanner desde la librería java.util. Al igual que en otros programas, esto permite leer datos introducidos por el usuario desde el teclado.
* **Definición de la clase y método principal:** El programa define una clase llamada ValidarHora y dentro de ella, el método main, que es el punto de inicio donde se ejecuta el código. El método main contiene toda la lógica para solicitar la hora y validarla.
* **Declaración de las variables H, M, y S:** Se declaran tres variables de tipo entero (int) llamadas H, M, y S. Estas representarán las horas, minutos y segundos, respectivamente. Las variables almacenarán los valores que el usuario introduzca para validar si se trata de una hora válida.

**Captura de los valores introducidos por el usuario:**

* El programa solicita al usuario que introduzca un valor para la hora (H) con System.out.print("Introduzca hora: ");, y lo almacena usando H = sc.nextInt();.
* Luego, hace lo mismo para los minutos (M) y los segundos (S), capturando los valores ingresados con sc.nextInt().

**Validación de la hora:** La validación de la hora ocurre con una sentencia condicional if. El programa verifica si los valores introducidos están dentro de los rangos aceptables para una hora válida:

* La hora (H) debe estar entre 0 y 23 inclusive.
* Los minutos (M) y los segundos (S) deben estar entre 0 y 59 inclusive.

Esto se realiza con la condición:  
if (H >= 0 && H < 24 && M >= 0 && M < 60 && S >= 0 && S < 60), que evalúa todas las condiciones a la vez. Si todas son verdaderas, entonces se imprime "Hora correcta".

**Mensaje de salida:**

* Si la hora es válida (cumple las condiciones), el programa muestra el mensaje "Hora correcta".
* Si no es válida, la condición else imprime "Hora incorrecta".

**Ejercicio10:** Programa que lea una variable entera mes y compruebe si el valor corresponde a un mes de 30 días, de 31 o de 28. Supondremos que febrero tiene 28 días. Se mostrará además el nombre del mes. Se debe comprobar que el valor introducido esté comprendido entre 1 y 12.

Texto

Descripción generada automáticamente

* **Importación de la librería java.util.Scanner:** La primera línea del código import java.util.\*; importa la librería Scanner, que se utiliza para capturar la entrada del usuario desde el teclado.
* **Definición de la clase CondicionalMes:** La clase CondicionalMes contiene el método principal main, que es el punto de inicio de la ejecución del programa.
* **Creación del objeto Scanner:** Dentro del método main, se crea un objeto Scanner llamado sc con la línea Scanner sc = new Scanner(System.in);. Este objeto será usado para leer la entrada del usuario.
* **Declaración de la variable mes:** Se declara la variable mes de tipo int, que almacenará el número que el usuario ingrese para representar el mes.
* **Solicitud del número de mes al usuario:** Con System.out.print("Introduzca número de mes: ");, el programa pide al usuario que introduzca un número. Luego, con mes = sc.nextInt();, el número ingresado se almacena en la variable mes.
* **Verificación de la validez del mes:** Antes de realizar cualquier acción, se valida si el número ingresado es válido. Si el número está fuera del rango 1 a 12, el programa imprime el mensaje "Mes incorrecto". Esto se maneja con el condicional: