

EJERCICIOS UNIDAD 2

Contenidos:

- Uso de operadores aritméticos, relacionales y lógicos.
- Estructuras condicionales.

Crea un nuevo proyecto independiente para cada uno de estos ejercicios. En todos los proyectos crea un comentario en el que aparezca tu nombre completo.

Realiza una captura de pantalla de tu solución y pégala en el documento.

! **Para la realización de estos ejercicios queda completamente prohibido utilizar ninguna herramienta externa, salvo la documentación oficial de Java.**

Ejercicio 1: Solicita por teclado un número de tipo int al usuario y escribe un programa que muestre true o false, dependiendo de si el número es positivo o no.

Ejercicio 2: Solicita la velocidad de un vehículo. Si supera los 120 km/h, muestra “Exceso de velocidad detectado”.

Ejercicio 3: Diseñar un programa que pida un número entero al usuario por teclado y a continuación muestre un mensaje indicando: “El número introducido es par” o “El número introducido es impar” o “El número introducido es el cero, no se si es par o impar”.

Requisito: Solo se pueden usar condicionales simple (sin la cláusula else).

Ejercicio 4: Crea una aplicación que le pida al usuario el precio de la compra total de los artículos que ha adquirido en una tienda. La aplicación le mostrará un mensaje “Debe abonar X euros”. Hay que tener en cuenta que si la compra que ha realizado el usuario es igual o superior a 500€ se le aplicará un 10% de descuento.

Ejercicio 5: Crea un programa que pida dos números tipo byte y devuelva si son iguales o no con un mensaje descriptivo como por ejemplo: “El número A y el número B son iguales”

Ejercicio 6: Crear una aplicación que pida dos números distintos y diga cual es mayor.

Ejercicio 7: Implementar un programa que pida por teclado un número decimal e indique si es un número casi-cero, que son aquellos, positivos o negativos, que se acercan a 0 por menos de una unidad, aunque curiosamente el cero no se considera un número casi-cero.

Ejemplos de números casi-cero son el 0,3, el -0,99 o el 0,123; algunos números que no se consideran casi-ceros son: el 12.3, el 0 o el -1.

Ejercicio 8: Realizar una aplicación que solicite al usuario la nota que ha conseguido en el módulo de programación. Si la nota es 5 o superior le mostrará un mensaje de “Enhorabuena has aprobado”, en caso contrario “Venga! Tienes que seguir intentándolo”.

Ejercicio 9: Escribir un programa que Una aplicación de conciertos quiere determinar si un usuario puede acceder a la zona VIP. Para poder entrar a la zona VIP se cumple la siguiente regla:

- Debe tener más de 18 años y disponer de entrada VIP,
- ó
- Ser menor de 18 pero hijo de un artista invitado (en este caso puede entrar aunque no tenga entrada VIP).

El programa debe pedir:

*La edad del usuario

*Si tiene entrada VIP (true/false)

*Si es hijo de un artista invitado (true/false)

Y mostrar:

“Acceso concedido a la zona VIP” si cumple la condición.

“Acceso denegado” en caso contrario.

Ejercicio 10a: Crea un programa en el que se pidan dos números reales al usuario y los muestre ordenados de menor a mayor. Para la realización de este ejercicio NO se puede utilizar las sentencias “if” ni “if-else”, debe realizarse utilizando el operador ternario.

Ejercicio 10b: Crea un programa que pida un número al usuario y muestre su valor absoluto. Para la realización de este ejercicio NO se puede utilizar las sentencias “if” ni “if-else” en caso de necesidad se utilizará el operador ternario.

Ejercicio 11: Copia y pega el código del ejercicio 6, y modifícalo para que el programa te indique qué número es mayor o si son iguales que le diga al usuario que ambos números son el mismo.

Para la realización de este ejercicio se debe utilizar las sentencias if-else.

Ejercicio 12: Crea un programa que pida al usuario 3 números enteros y los muestre ordenados de mayor a menor. Vamos a suponer que los 3 números que ingresa el usuario son distintos.

Ejercicio 13: Un alumno de bachillerato está cansado de tener que hacer los cálculos de las ecuaciones de segundo grado una y otra vez en sus clases de matemáticas. Para ello se va a hacer un programa en Java que pida los coeficientes a, b y c y le muestre los resultados reales en el supuesto de que existan. Si no se puede calcular también se lo va a indicar indicándole el error.

Hay que tener en cuenta que las soluciones de una ecuación de segundo grado, $ax^2+bx+c=0$, son:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

AYUDAS:

- La clase Math te puede ayudar a resolver raíces cuadradas.
- Hay algunas operaciones que no se pueden realizar como dividir entre cero.
- Hay otra más en esta fórmula que tienes que tener cuenta.

Ejercicio 14: Escribir una aplicación que reciba un número entre 0 y 99999 (ambos incluidos) e indique el número de cifras que tiene. Por ejemplo si introducimos el número 7 nos indica que tiene una cifra, sin embargo, si introducimos el 2025 nos indica que tiene 4 cifras.

ATENCIÓN: Ahora ya podemos realizar alguna comprobación para que los datos

que introduce el usuario se vayan ajustando al enunciado. Se debe incluir la comprobación de que si el número que introduce no está entre 0 y 99999 la aplicación indique “Número inválido: no comprendido entre 0 y 99999” y finalice su ejecución.

Ejercicio 15: Modifica el ejercicio anterior para que los números que pueda introducir el usuario sea entre -99999 y 99999 ambos incluidos.

Ejercicio 16: Crea un programa que le pida al usuario un número del 1 al 7 y el programa muestre el día de la semana que es, por ejemplo, si el usuario introduce el número 1 el programa muestra “lunes”, si el usuario introduce el 7 muestra “domingo”.

NOTA: Se debe utilizar la estructura del switch para resolver el ejercicio.

Ejercicio 17: Crea un programa que muestre un menú con varios superhéroes (1. Spiderman, 2. Batman, 3. Wonder Woman, 4. Iron Man).

El usuario introduce un número del 1 al 4 y el programa responde con una frase motivadora sobre ese superhéroe.

(Ejemplo: “¡Eres ágil como Spiderman!”)

Ejercicio 18: Crea un programa que pida un número (puede tener decimales) al usuario. A continuación, muestra el siguiente menú:

Pulsa 1 para calcular la raíz cuadrada del valor introducido.

Pulsa 2 para redondear hacia arriba

Pulsa 3 para redondear hacia abajo

Pulsa 4 para calcular el logaritmo natural

Dependiendo de la opción pulsada el programa realiza los cálculos correspondientes y le muestra un resultado.

Ejercicio 19: Crea un programa que le pida al usuario el día, mes y año de una fecha y el programa debe indicar si la fecha es correcta o no. Hay que tener en cuenta que existen meses con 28,30 y 31 días. No se tienen en cuenta los años bisiestos.

Ejercicio 20: Vamos a crear un conversor de euros a diferentes monedas. La aplicación le pedirá al usuario la cantidad en euros y mostrará un menú para que el usuario decida si quiere convertirlo a: Dólar estadounidense (USD), Libra esterlina (GBP), Yen japonés (JPY), Peso mexicano (MXN) o Franco suizo (CHF).

Las tasas de cambio son:

1 € = 1,1739 USD

1 € = 0,9347 CHF

1 € = 21,64 MXN

1 € = 173,79 JPY

1 € = 0,87315 GBP

NOTA: Este ejercicio debe realizarse utilizando la nueva sintaxis de la estructura switch

Ejercicio 21: Usando la estructura switch como expresión que devuelve un valor, haz un programa que le pida un número entre 1 y 12 al usuario. Estos números se corresponden con los meses del año (enero, febrero, marzo...). El programa devolverá por pantalla el número de días que tiene ese mes.

(podemos suponer que febrero tiene siempre 28 días)

ACTIVIDADES EXTRAS DEL TEMA

Para resolver estas actividades solo se pueden utilizar las estructuras de control explicadas hasta la fecha.

Ejercicio 22: Crea una aplicación que solicite al usuario una fecha (día, mes y año) y muestre la fecha correspondiente del día siguiente.

Recuerda que no todos los meses tienen el mismo número de días. Para la resolución de este ejercicio no tendremos en cuenta los años bisiestos, por lo que febrero siempre tendrá 28 días.

También vamos a suponer que el usuario introduce datos válidos, no hace falta realizar esas comprobaciones.

Ejercicio 23: Escribir un programa que calcule el dinero recaudado en un concierto. La aplicación solicitará el aforo máximo del local, el precio por entrada

(suponemos que todas las entradas tienen el mismo precio) y el número de entradas vendidas. Hay que tener en cuenta que si el número de entradas vendidas no supera el 20% del aforo del local, se cancela el concierto. Si el número de entradas vendidas no supera el 50% del aforo del local, se realiza una rebaja del 25% del precio de la entrada.

🔥 **Ejercicio 24:** Escribe una aplicación que solicite al usuario un número comprendido entre 0 y 9999. La aplicación tendrá que indicar si el número introducido es capicúa.

Ejercicio 25: Escribe una aplicación que solicite por consola dos números reales que corresponden a la base y la altura de un triángulo. Deberá mostrarse su área, comprobando que los números introducidos por el usuario no son negativos, algo que no tendría sentido.

Ejercicio 26: Utiliza el operador ternario para calcular el valor absoluto de un número que se solicita al usuario por teclado.

Ejercicio 27: El DNI consta de un entero de 8 dígitos seguido de una letra que se obtiene a partir del número de la siguiente forma:

letra = número DNI módulo 23

Basándote en esta información, elige la letra a partir de la numeración de la siguiente tabla:

Resto 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22	Letra T R W A G M Y F P D X B N J Z S Q V H L C K E
--	---

Diseña una aplicación que solicite un DNI y muestre el DNI con su letra.