Nombre: _____

Tarea presencial
Unidades 1 y 2.

Indicaciones de entrega

Una vez realizada la tarea, el envío se realizará a través de la plataforma Moodle, en la tarea "Tarea presencial 1. Introducción y estructuras de control".

Debes comprimir en un archivo con extensión .zip **ÚNICAMENTE los 3 archivos .java** que necesitas para resolver los problemas. Nombra dicho archivo de la siguiente forma *Apellido1_Apellido2_Nombre_Tarea_Presencial_1.zip*

Evaluación

Cómo se valora y puntúa la tarea

Ejercicio 1 → 2,25 puntos
Ejercicio 2 → 2,25 puntos
Ejercicio 3 → 2,25 puntos

Cuestión 4 → 0,4 puntos
Cuestión 5 → 0,45 puntos
Cuestión 6 → 0,4 puntos
Cuestión 7 → 0,4 puntos

Cuestión 8 → 0,4 puntos
Cuestión 9 → 0,4 puntos
Cuestión 10 → 0,4 puntos
Cuestión 11 → 0,4 puntos

***Cada cuestión incorrecta resta 0,15 puntos**

Factores que influyen negativamente en la puntuación de cada ejercicio

- El código no compila y/o no se ejecuta correctamente
- Errores de base sobre conceptos de temas anteriores: nombres de identificadores incorrectos, elección de tipo inadecuado, etc.
- Elección de la estructura de control de flujo más apropiada para cada solución: if-else, switch-case, for, while...
- Uso de procedimientos, técnicas, estructuras de datos, funcionalidades, etc. No vistas en clase hasta el momento
- Las soluciones no se adaptan al enunciado, ya sea por quedarse cortas o excederse del mismo.
- No mostrar mensajes indicativos de la acción correspondiente, ya sean para datos de entrada o para mostrar resultados.

Resultados de aprendizaje que se evalúan en la tarea

RA1: Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
RA3: Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

Indicaciones adicionales

Entrada de datos por consola

```
// Creación del objeto Scanner
Scanner scan = new Scanner(System.in);
// Uso del Scanner para distintos tipos
scan.nextInt(); //Captura un entero
scan.nextDouble(); // Captura un número con decimales
scan.next(); // Captura la línea escrita hasta darle al intro
scan.nextLine(); // Captura la línea escrita hasta darle al intro
```

Salida de datos por consola

```
// Imprime texto por la consola
System.out.println("Cadena de texto");
// Imprime la concatenación de las cadenas pasadas por parámetro
System.out.println("Cadena de texto" + "otra cadena");
```

Estructuras de control válidas

1. If
2. If – else
3. If – else if
4. If – else if - … – else
5. Switch – case
6. For
7. While
8. Do – while

Tipos aceptados

1. Tipos primitivos: int, long, float, double, char, boolean…
2. Cadenas de caracteres (String).
   Principales operaciones con cadenas:
   - Creación: String cadena = “hola”;
   - Obtener longitud: cadena.length()
   - Concatenación: cadena1 + cadena2 / cadena1.concat(cadena2)
   - Comparación: cadena1.equals(cadena2) / cadena1.equalsIgnoreCase(cadena2)
   - Obtener subcadenas: cadena.substring(1, 3)
   - Operaciones de may/min: cadena.toUpperCase() / cadena.toLowerCase()
   - Conversiones de tipo: String.valueOf(otroTipoDeDato)

Ejercicios

1. Realizar un programa que muestre en pantalla el siguiente menú:

```
------------------| MENÚ PRINCIPAL |--------------
| 1 – Módulo de Programación                      |
| 2 – Módulo de Lenguaje de Marcas                |
| 0 – Resumen y salida del programa               |
--------------------------------------------------
```

- Posteriormente se debe quedar a la espera de que el usuario introduzca un valor numérico.
- Si se introduce la opción 1 o 2, solicitará un valor con decimales. Dicho valor será guardado en la nota del módulo correspondiente.
  (0,25) EXTRA: controla que la nota tenga un valor comprendido entre 0 y 10, ambos incluidos.
- Si introduce un 0 el programa mostrará un mensaje mostrando las calificaciones de cada uno de los módulos y finalizará. Si no se ha introducido valor para alguna nota se mostrará "NOEV"
  - Sin nota introducida ---> NOEV

  Ejemplo:

```
Calificaciones DAW
Módulo de Programación: 5,7
Módulo de Lenguaje de Marcas: NOEV
```

- Si se introduce cualquier otra opción distinta a 0, 1 o 2, el programa mostrará un mensaje de error y volverá al menú.

2. Mostrar por pantalla ordenados descendentemente los números enteros positivos menores que 100 de tal forma que vayan 4 impares seguidos de un número par.
Se comenzará por el número 100.
De la siguiente forma: 100, 99, 97, 95, 93, 92, 91, 89, 87, 85, 84, 83, 81,79, 77…

3. Una empresa quiere actualizar los sueldos de los empleados de la siguiente forma:
0 a 9000 Euros, aumento un 20%
9001 a 15000 Euros, aumento 10%
15001 a 20000 Euros, aumento 5 %
Más de 20000 Euros, aumento 1 %.
Escriba un programa que permita tantas veces como se desee introduciendo un sueldo:

- Mostrar el sueldo actual
- El incremento en Euros
- El nuevo sueldo a cobrar
  (0,25) EXTRA: Una pregunta al usuario si quiere salir del programa. Si introduce los caracteres “s” o “S” saldrá del programa, en cualquier otro caso, se pedirá otro sueldo para actualizar (**Nota:** no solo con la “N” o “n”).

```
Sueldo actual: 17000
Incremento: 850
Nuevo sueldo = 17850
¿Desea salir S/N?
```

Nombre: _____

Tiempo para realizar las cuestiones 20 minutos.

4. Dado el siguiente fragmento de código:

```
int acumulador=1;
for (int contador=1; contador<=4; contador++) {
  acumulador = acumulador * contador;
}
```

Calcular cuánto valdrá la variable acumulador tras la ejecución del bucle.

a. Existe un error de compilación.
b. acumulador valdrá 10.
c. acumulador valdrá 12.
d. acumulador valdrá 24.

5. Dado el siguiente fragmento de código:

```
int numero = 7;
int contador;
int resultado=0;
contador = 1;
do {
      resultado = contador * numero;
      System.out.println (numero + " x " + contador++ + " = " +
resultado);
} while (contador < 10);
```

Indicar cuáles de las siguientes afirmaciones son correctas. Seleccione una o más de una:

a. El valor final de la variable contador es 9.
b. El código dentro del do se ejecuta nueve veces.
c. Se trata de un bucle infinito que jamás finalizará su ejecución.
d. El valor final de la variable contador es 10.
e. El valor final de la variable resultado es 70.
f. El valor final de la variable resultado es 63.

6. Si quisieras implementar en un programa Java un menú en el que hubiera que seleccionar entre varias opciones, ¿qué estructura utilizarías para distinguir entre una u otra de las opciones elegidas?

a. If-else
b. break
c. select
d. switch

7. La estructura do-*while* se considera un bucle controlado por _____.
a. contador
b. sucesos
c. variables
d. aritmética

8. El uso en Java de etiquetas para indicar posiciones de código a las que saltar usando las sentencias `break` y `continue`…
a. es muy aconsejable, ya que nos aporta flexibilidad, al permitirnos evitar las rígidas reglas de las estructuras de control de flujo.
b. es desaconsejable ya que introduce una complejidad que siempre es evitable.
c. es una característica destacada de los lenguajes de programación orientados a objetos.
d. es una ventaja en comparación con otros lenguajes como Python o PHP que no las tienen.

9. En aquellos casos en los que las instrucciones que forman el cuerpo del bucle necesitan ser ejecutadas al menos una vez, es muy útil la estructura de repetición…
a. For-each
b. While
c. Do-while
d. Switch

10. Qué se imprimirá por consola tras ejecutar este fragmento de código:

```
String opTernario = "XD";
opTernario = (2 == 4%2) ? "Qué tal" : (4 >= 7/3+2) ? "Hola" : "Adiós";
System.out.println(opTernario);
```

a. XD
b. Qué tal
c. Hola
d. Adiós

11. Dado el siguiente fragmento de código:

```
int a=1, b=2, auxiliar=0;
b=a;
a=b;
```

Calcular el valor de las variables a y b tras su ejecución.
a. Tanto a como b valdrán 2.
b. Error de compilación.
c. Tanto a como b valdrán 1.
d. a valdrá 2 y b valdrá 1.