|  |  |  |
| --- | --- | --- |
| **PROGRAMACIÓN – 1º D.A.W. IES DOMENICO SCARLATTI**  **EXAMEN TIPO 2ª EVALUACIÓN – PARTE 2** | | Tipo: Sim. |
| APELLIDOS: | Calificación: | |
| NOMBRE: |
| FECHA: 02-2021 (simulacro) |

**PUNTUACIÓN: 5 puntos. TIEMPO: 70 minutos**  
**Importante:** indenta (sangra) el código para mejorar su legibilidad. No sangrar el código supone una penalización de -0’25 en cada ejercicio.

1. La clase Empleado almacena la siguiente información de cada empleado de una empresa:  
   1. Nombre y apellidos.
   2. Fecha de alta en la empresa.
   3. DNI.
   4. Salario
   5. Número de empleado.

Codifica la clase Empleado con métodos set y get (solo se puede acceder a los atributos desde otras clases con estos métodos). El número de empleado se asignará automáticamente al crear un empleado.  
El constructor de la clase Empleado recibirá como parámetro el DNI. En el constructor se inicializará también el campo fecha de alta tomando la fecha actual. ***(0’75 puntos)***

1. Codifica un programa que lea diez palabras o frases por consola y las almacene en un array. A continuación el programa las imprimirá todas en orden inverso: primero la última leída y, la última, la primera que se leyó.   
     
   Además, el programa debe escribir qué palabra es la “menor” de todas: la que iría antes en un diccionario sin distinguir entre mayúsculas y minúsculas. ***(1 punto)***
2. La clase Animal (de la que no se pueden crear objetos) tiene un atributo denominado peso (valor entero).  
   La clase Mamífero hereda de animal y tiene el atributo ritmoCardiaco (entero). Las clases Perro, Gato y Vaca heredan de Mamífero y añaden el atributo edad (entero) que es privado, pero que puede consultarse. Las clases Perro y Gato implementan el interfaz Mascota que tiene solo un método: jugar().   
     
   Implementa todas las clases anteriores y crea una clase, ReinoAnimal, que cree dos animales de cada tipo, con un peso y un ritmo cardiaco que tú les des y los almacene en un array denominado arca. Debes a continuación recorrer el array y hallar la suma total de los **pesos** de todos los animales que has almacenado en el array arca, así como el **ritmo cardiaco** medio.   
    ***(1’25 puntos)***
3. Codifica el método public boolean isInside(String contenedor, String seq) que comprueba si los caracteres de seq aparecen, en orden, dentro de la cadena contenedor (no necesariamente consecutivos).  
   Es decir: si contenedor fuese “Hola a todos” y seg fuese “ latd”, el método devolvería “true” porque “latd” aparece (con otros caracteres intercalados) en la cadena contenedor.  
   Si seq fuese “dalo” debería devolver “false”. ***(1 punto)***

1. Codifica un método que te calcule el factorial de un número de forma **recursiva**. Si el número pasado es mayor que 15 se devolverá -1.   
   ***(0’5 puntos)***
2. ¿Qué calcula el siguiente método? ¿Qué se imprimiría por pantalla si se le pasa el valor 6?  
   ***(0’5 puntos)***

public static int enigma (int n){

int resul = 0;

if (n == 1) {

resul = 1;

}

else {

resul = n + enigma(n - 1);  
 System.out.println(resul);

}

return resul;

}

=============================================

**EJERCICIOS SOBRE ARRAYS**

1. Crear una tabla bidimensional de tamaño 5x5 y rellenarla de la siguiente forma: la posición T[n,m] debe contener n+m. Después se debe mostrar su contenido.
2. Crear y asignar valores a una tabla de tamaño 4x4 y decir si es simétrica o no, es decir si se obtiene la misma tabla al cambiar las filas por columnas.
3. Codifica el método **sumaMatrices** que reciba dos arrays bidimensionales de número enteros, ambos del mismo tamaño y el número de filas y columnas y devuelva un array con la suma de ambos arrays.  
   (mirar cómo es la suma de dos matrices en: <https://www.matesfacil.com/matrices/resueltos-matrices-suma.html>)
4. Haz código para probar el método sumaMatrices. Imprime el resultado devuelto.
5. Realiza el método imprimeMatrizTranspuesta(int[][] matriz, int filas, int columnas) que imprime la matriz transpuesta de la que hemos recibido (consulta la referencia anterior para ver qué es la matriz transpuesta si no lo recuerdas).
6. Codifica y prueba el método copiaArray() que recibe un array de enteros y devuelve una copia (nuevo objeto) del array. Realiza código para probarlo.
7. Codifica y prueba el método eliminaDuplicados() que recibe un array de enteros sin ordenar y devuelve un array sin números duplicados. El tamaño del array devuelto debe ajustarse a la cantidad de números resultantes.

Realiza código para probarlo.

1. Codifica el método rotarDerecha(int[] array, int posiciones) que desplaza el contenido del array a la derecha según el número de posiciones especificado.  
   Ejemplo:  
   int[] n = {1, 2, 3, 4}; rotarDerecha(n, 2);  
   El contenido ahora es: 3, 4,1,2  
   Realiza código para probarlo.

.