

## PROGRAMACIÓN - EXAMEN FINAL 2º EVALUACIÓN

C.F.G.S 1º D.A.M.

12-Marzo-2021

I.E.S. Fernando Aguilar Quignón

Al entregar tu examen, debes ser capaz de explicar cada parte de tu código. No está permitida la copia de compañeros u otro medio. No puedes comunicarte con tus compañeros de ninguna manera mientras el examen está en progreso. En caso de detectar copias, el examen se puntuará con un 0.

### Ejercicio 1 - RA6 Tipos de datos avanzados - 3,5 puntos.

Crea una matriz de X filas  $\times$  Y columnas y Z números aleatorios sin repetirse. El programa se debe componer de los siguientes cinco métodos estáticos:

**private static int comprobarAleatorio (int numero) (0,5 puntos)** // La clase principal debe contar con el vector int [] aleatorio. Dentro de este método comprobarAleatorio tendremos que poner a 0 cada posición del vector que coincida con cada número random generado. De esta forma si dicha posición tiene un 0 sabremos que el número ya ha sido introducido en la matriz. Si el número no ha sido introducido, el método devolverá el mismo parámetro de entrada, es decir, número.

**private static void rellenarVectorAleatorio (int longitud) (0,5 puntos)** // Método donde tenemos que inicializar el atributo int[] aleatorio con su debido tamaño y rellenar todos sus valores.

**private static void rellenarComprobarRepetidos (int[][] miMatriz, int numGenerar) (0,75 puntos)** // Método que recibe como parámetro una matriz del número de filas y columnas indicadas por el usuario, y el valor máximo de números aleatorios a generar. En el interior del método, generaremos los números aleatorios desde el valor 1 en adelante, se llamará al primer método comprobarAleatorio para hacer las comprobaciones oportunas, y rellenaremos la matriz.

**private static void visualizarMatriz (int[][] miMatriz) (0,75 puntos)** // Este método recibe la matriz principal ya rellena con todos sus valores aleatorios distintos, y la imprime por pantalla con sus números en color verde.

**public static void main (String[] args) (0,75 puntos)** // El método main deberá llamar a los principales métodos vistos y generar una salida por pantalla como la siguiente:

```
Matriz de X filas x Y columnas rellena de Z números aleatorios sin repetir

Introduzca valor entero positivo de la "fila": 6
Introduzca valor entero positivo de la "columna": 8
El primer número aleatorio utilizado será el valor 1
¿Cuál es el número máximo de valores aleatorios sin repetir que deseas generar?: 48
16      25      43      8      26      46      5      18
 2       7      35      33      45      48      29      17
13      12      21      27      34      47      15      32
42      11      14       9      20      38      44       6
23      31      28      39       3      22      19      10
 1      40       4      37      24      41      36      30
BUILD SUCCESSFUL (total time: 8 seconds)
```

Además, dentro del método main, en el caso de que el número de celdas sea mayor que la cantidad de números aleatorios a rellenar sin repetir, se debe mostrar un mensaje de error al usuario y pedir todos los datos de nuevo. (0,25 puntos)

## Ejercicio 2 - RA6 Tipos de datos avanzados - 3 puntos.

Crea un programa que nos permita administrar los datos de los trabajadores de una empresa. Para ello, implementa la estructura “trabajadores” compuesta de NIF (clave) y nombre (valor) de trabajadores.

El programa tendrá un menú con las siguientes opciones:

- 1.- Nuevo trabajador: El sistema pedirá al usuario el NIF del nuevo trabajador. Comprobará si el NIF ya estaba introducido antes, en caso de no estarlo, pedirá el nombre y añadirá ambos valores, y en caso de estar en la estructura mostrará un mensaje de error. (0,25 puntos)
- 2.- Buscar trabajador por NIF: Si el NIF introducido por el usuario existe, mostramos su nombre, en caso contrario mostramos un mensaje de error. (0,25 puntos)
- 3.- Buscar trabajador por nombre: Si el nombre introducido por el usuario existe, mostramos su NIF, si no existe, mostramos un mensaje de error. Para resolver este apartado utiliza un iterador (0,75 puntos)
- 4.- Modificar trabajador: El usuario introduce el NIF del trabajador, se comprobará si ese NIF existe. Si existe, se borra el trabajador, se pide el nuevo nombre, y se crea otro con los nuevos datos. En caso de que no exista el NIF introducido por el usuario, se muestra un mensaje de error. (0,75 puntos)
- 5.- Mostrar datos de todos los trabajadores: Utiliza un for extendido para recorrer todos los trabajadores de la empresa y mostrar su NIF y nombre. (0,75 puntos)
- 6.- De modo teórico, explica para que sirven las clases y métodos genéricos. Escribe un ejemplo. (Realiza este apartado como comentarios del fichero .Java de este ejercicio). (0,25)

El método main deberá generar una salida por consola como la siguiente:

```
1.- Nuevo trabajador.
2.- Buscar trabajador por NIF
3.- Buscar trabajador por nombre
4.- Modificar trabajador
5.- Mostrar datos de todos los trabajadores
Elija una opcion (0 para salir):
```

### Ejercicio 3 - RA5 E/S - 2 puntos.

Desarrolla una aplicación que pida al usuario una cadena a buscar y el nombre de un fichero. A continuación, mostrará cuantas veces aparece esta palabra en el fichero y en que líneas aparece. La salida por pantalla debe ser algo similar a lo siguiente:

```
Introduce la cadena a buscar:
viaje
Introduce el nombre del fichero:
examen.txt
Líneas donde aparece: [2, 3, 4]
BUILD SUCCESSFUL (total time: 12 seconds)
```

Nota: Para simplificar el proceso de indicar el nombre del fichero, crea dicho fichero en el mismo directorio que el proyecto y así no tener que indicar la ruta completa.

### Ejercicio 4 - RA5 E/S - 1,5 puntos.

Crea una pequeña calculadora que tenga una interfaz gráfica de usuario igual a la de las siguientes imágenes. La calculadora solo realizará sumas y restas, y sus componentes serán etiquetas, cuadro de textos, botones y radio botones excluyentes entre sí. El usuario escribirá los valores en los cuadros de texto, y el resultado se mostrará en una etiqueta cuando se pulse el botón “Calcular”.

