



Programación

Unidad 3: Estructuras de datos estáticas y dinámicas

Realizar los programas siguientes en lenguaje JAVA. A Moodle subirás un fichero zip con el código del fichero java correspondiente o el enlace Github.

1. (3 puntos) Para las oposiciones de informática se van a sacar 5 bolas de entre 65 temas a estudiar. El opositor se ha estudiado los temas: 4, 11, 23, 34, 35, 36, 39, 50, 60 y 65. En un menú tendrás las siguientes acciones:

1. Generar un vector aleatorio donde asegures que no se repita ningún número. Ejemplo: {1,60,44,12,23} (1,5 puntos)
2. Realiza un función que valide el vector generado con el vector del opositor y devuelva los temas que coinciden **ordenados**. Si hay coincidencias dile enhorabuena y si no, que lo sientes. En el ejemplo, dirá: “Temas coincidentes: 23, 60. Enhorabuena” (1,5 puntos)

Elabora este programa utilizando la clase `ArrayList<Integer>`.

2. (2 puntos) Implementa la función `obtenerVector(array, num)`, donde obtengas como salida un array que en cada posición tiene la suma de los num elementos consecutivos. Tienes que verificar que num es un número menor que el tamaño del vector. Ejemplo para un array que tiene {10,1,5,8,9,2} y num=3, el resultado saldría {16, 14, 22, 19}, donde 16= (10+1+5), 14=(1+5+8), 22=(5+8+9) y 19=(8+9+2). El array de salida se genera como array dinámico.

3. (3 puntos) Escribe un programa que trabaje con una matriz de enteros de tamaño 5×5 . El programa a través de un menú debe:

1. Rellenar la matriz automáticamente con números enteros aleatorios entre 1 y 50 pero teniendo presente que en cada fila serán los rangos: {1-10}, {11-20}, {21-30}, {31-40}, {41-50}. En este caso, harás una función `generaFila(min,max)` y la asignarás a cada fila de la matriz 5x5. (1 punto)
2. Mostrar la matriz en formato de tabla. (0,25 puntos)
3. Calcular y mostrar los siguientes valores sobre la matriz: (1,25 puntos)
 - La **suma de cada fila**.
 - La **suma de cada columna**.
 - El **valor máximo y mínimo** de toda la matriz y su posición (fila y columna).
4. Modifica el valor de una coordenada introducida por el usuario. Hay que verificar que no se introducen valores fuera de sus límites. (0,5 puntos)

4. (2 puntos) Utilizando las funciones de String, crea la función **verificaAnagrama** que determina si dos palabras introducidas por teclado, una es un anagrama de la otra, es decir, contiene las mismas letras pero cambiadas de orden. Tendrás que pasarlas a `char[]`, ordenarlas y ver si son iguales. Además, deberás pasar todas las letras a minúscula y antes de comprobar si es anagrama ver los tamaños de ambas palabras. Ejemplos: Amor-roma o nido-oDin son anagrama pero alumno-clase no, por tener tamaño diferentes, ni nido-Oden porque la i no está en Oden.