

Tema 3 Arrays y colecciones

Instrucciones con acceso a celdas

1. Hágase un programa en el que se declare un array de reales de tamaño 3. En el programa se hará lo siguiente;
 - a) Se introducirán los datos del array por la consola.
 - b) Si el valor de la segunda celda es mayor que el de la primera celda, se incrementará en un uno el valor de la tercera celda; si no, se le asignará a la tercera celda un valor aleatorio real entre 0 y 1 (este último sin incluir).
 - c) Se mostrarán los valores de las tres celdas en una misma línea separados por comas.
 - d) Se generará una posición aleatoria del array (un valor entre 0 y 2, ambos inclusive). La celda aleatoria generada se le asignará como valor la suma de las otras dos celdas.
 - e) Se mostrará de nuevo las celdas con el formato del apartado c)
 - f) Se reinicializará el array a uno de tamaño 2 con valores aleatorios de números enteros entre 0 y 9, ambos inclusive.
 - g) Se mostrará el contenido de las dos celdas con el siguiente formato:

```
| --- | --- |  
|  5 |  0 |  
| --- | --- |
```

(Se observa que los valores reales de las celdas a mostrar se han truncado a entero)

- h) Se intercambiarán el contenido de las dos celdas
- i) Se mostrará de nuevo las celdas con el formato siguiente:

0	5
---	---

(AccesoCeldas)

Recorrer un array

2. Se considera un array con 10 enteros con valores aleatorios entre 0 y 9, ambos inclusive. Se mostrarán los datos separados por una coma y un espacio. Se modificará el array de la manera indicada en cada apartado, y se mostrará de nuevo.
 - a) Incrementar en 1 los valores pares y en -1 los impares.
 - b) Duplicar los valores positivos menores que 5
 - c) Sumar a cada valor un valor entero aleatorio entre -5 y 5.

(RecorrerArray)

Recorrer parcialmente un array

3. Se considera un array de enteros cuyo tamaño se introduce por consola y se rellena con valores aleatorios entre 0 y 9, ambos inclusive. Se mostrarán los datos separados por una coma y un espacio. Se modificará el array de la manera indicada en cada apartado, y se mostrará de nuevo.
 - a) Mover los datos una posición hacia la derecha (el primero pasa al segundo, el segundo al tercero, ..., y el último al primero)
 - b) Invertir el array
 - c) Mostrar la posición del primer par y del último impar.

(RecorrerParcialmenteArray)

Recorrer un array con saltos

4. Se considera un array de cadenas cuyo tamaño se introduce por consola. Se controla que el número de elementos del array sea par. Se leen por consola cada uno de los valores del array. En el programa se intercambiarán las cadenas de la primera celda y la segunda; la tercera y la cuarta; y así sucesivamente hasta intercambiar las cadenas de la penúltima celda y la última celda. Por último, se mostrarán las cadenas de cada celda en líneas distintas y enumeradas.

(RecorrerArraySaltos)

Métodos de arrays

5. Introdúzcase, una a una, 10 temperaturas entre los valores de -10.0 y 50.0. A medida que se introducen se mostrarán todas, se obtendrán la menor y la mayor temperatura, la media de las temperaturas con dos decimales, y la media con dos decimales sin tener en cuenta las temperaturas máximas ni mínimas (puede ser que no se pueda obtener dicha media). Se tendrán los siguientes métodos:
 - leer (temperatura validándola)
 - rellenar (temperaturas)
 - mostrar (todas las temperaturas)
 - mayor (temperatura)
 - menor (temperatura)
 - media (de todas las temperaturas)
 - mediaSin (de todas las temperaturas quitando los extremos)
 - mostrar media de las temperaturas
 - mostrar la media de las temperaturas sin la menor y mayor.

(TemperaturasArray)

6. Simúlese el juego de la carrera de caballos introduciendo el número de caballos que van a participar (como mínimo 2). Se tendrá un array que almacena el avance de cada uno de los caballos. Se definirán los siguientes métodos:
 - Leer el número de caballos

- Generar el caballo que avanza.
 - Avanzar caballo
 - Mostrar todos los caballos (limpiando previamente la pantalla)
 - Detener el juego durante 1 segundo.
 - Finalización del juego
 - Mostrar el caballo ganador.
(CarreraCaballosArray)
7. Se generará un array de 20 números reales con valores aleatorios entre 0.0 y 10.0, los cuales se mostrarán en líneas distintas con la posición que ocupan. Se obtendrá la suma de los números reales del array desde y hasta posiciones introducidas. (SumaDesdeHasta)
8. Considérese un array de enteros con valores aleatorios entre 0 y 99, que se mostrarán separados por un espacio. Se obtendrá el o los valores que más veces se repitan separados por un espacio con un texto del tipo “Los valores que más se repiten son” o “El valor que más veces se repite es el”. (MasSeRepite)
9. La criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores que un número natural dado n. Se forma una lista con todos los números naturales comprendidos entre 2 y n, y se van tachando los números que no son primos de la siguiente manera: Comenzando por el 2, se tachan todos sus múltiplos; comenzando de nuevo, cuando se encuentra un número entero que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos, así sucesivamente. El proceso termina cuando el cuadrado del mayor número confirmado como primo es mayor que n.

Determinemos, mediante el siguiente ejemplo, el proceso para determinar la lista de los números primos menores de 20.

Primer paso: listar los números naturales comprendidos entre 2 y 20.

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

Segundo paso: Se toma el primer número no rayado ni marcado, como número primo.

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

Tercer paso: Se tachan todos los múltiplos del número que se acaba de indicar como primo.

2	3	-	5	-	7	-	9	-	11	-	13	-	15	-	17	-	19	-
---	---	---	---	---	---	---	---	---	----	---	----	---	----	---	----	---	----	---

Cuarto paso: Si el cuadrado del primer número que no ha sido rayado ni marcado es inferior a 20, entonces se repite el segundo paso. Si no, el algoritmo termina, y todos los enteros no tachados son declarados primos.

Como $3^2 = 9 < 20$, se vuelve al segundo paso:

2	3	-	5	-	7	-	9	-	11	-	13	-	15	-	17	-	19	-
---	---	---	---	---	---	---	---	---	----	---	----	---	----	---	----	---	----	---

Se hace el tercer paso:

2	3	-	5	-	7	-	-	-	11	-	13	-	-	-	17	-	19	-
---	---	---	---	---	---	---	---	---	----	---	----	---	---	---	----	---	----	---

En el cuarto paso, el primer número que no ha sido tachado ni marcado es 5. Como su cuadrado es mayor que 20, el algoritmo termina y se consideran primos todos los números que no han sido tachados.

Como resultado se obtienen los números primos comprendidos entre 2 y 20, y estos son: 2, 3, 5, 7, 11, 13, 17, 19

Prográmesse la criba de Erastones de números naturales comprendidos entre 2 y un número natural que se introduce (entero largo mayor o igual que 2) a medida que se introduce el número (de ser vacío, 0 o 1, se mostrará como salida la cadena vacía) (CribaDeErastones)

10. Considérese el experimento de lanzar un dado “n” veces. En el experimento se mostrará el número de veces que se ha obtenido cada resultado y su probabilidad (cociente entre el número de veces que aparece dicho resultado y el total de tiradas) con 8 decimales. Modifíquese el programa para que el dado tenga como posibles resultados el 1, 2, 3, ..., “m” siendo “m” un valor que se introduce. Inicialmente “m” valdrá 6 y “n” 1000. (DadoMultiple)

Tipos básicos y de cadena por referencia

11. Rescribese el siguiente programa utilizando tipos básicos y de cadena por referencia

```
import fsg.in;
class TiposBasicosPorReferencia
{
    public static void main(String[] args)
    {
        int n1 = 0, n2 = 0;
        int sum = 0;

        n1 = in.leerInt();
        n2 = in.leerInt();
```

```
        if(n1%2==0) n1++;  
        else n2+=n1;  
  
        mostrar(n1);  
        mostrar(n2);  
        sum = sumar(n1,n2);  
        mostrar(sum);  
    }  
  
    static int sumar(int a, int b)  
    {  
        return a+b;  
    }  
  
    static void mostrar(int a)  
    {  
        System.out.println(a);  
    }  
}  
(TiposBasicosPorReferencia)
```

Parámetros por referencia de tipos básicos

12. Hágase un programa que lea dos números reales y dos palabras. El programa intercambiará los datos reales y las cadenas. Se volverán a mostrar los datos en el orden que fueron leídos. Defínanse y utilícese métodos para leer los datos, mostrarlos e intercambiarlos.

- Procedimiento con un parámetro por referencia de tipo real que permite leer un real y almacenarlo en el parámetro.
- Procedimiento con un parámetro por referencia de tipo cadena que permite leer una cadena y almacenarlo en el parámetro.
- Procedimiento con un parámetro por valor de tipo real que muestra el valor del parámetro por la consola
- Procedimiento con dos parámetros por referencia de tipo real que intercambia el valor contenido en dichos parámetros
- Procedimiento con dos parámetros por referencia de tipo cadena que intercambia el valor contenido en dichos parámetros

Arrays dinámicos

13. Hágase un programa que lea enteros hasta introducir un numero par. El programa mostrará todos los enteros introducidos que están entre los valores del primero y del último introducido, ambos inclusive. (ArrayDinamico)

Por ejemplo, si se introducen los números:

29 23 11 17 -11 12

La salida será: 29 23 17 12

14. Hágase un menú que dado inicialmente un array de enteros con 0 componentes, se tengan opciones para:

- a) Añadir un nuevo entero por el final.
- b) Añadir un nuevo entero en una posición
- c) Mostrar los valores contenidos separados por una coma.
- d) Eliminar la última posición.
- e) Eliminar una determinada celda.
- f) Leer un entero y eliminar todas las celdas que contienen dicho valor.
- g) Vaciar el array.

(MenuArray)

Procesamiento de múltiples arrays

15. Se considera un array de 10 elementos con valores entre 0 y 99 aleatorios, y otro array de las mismas características. Se mostrarán los dos arrays. Se podrán obtener los siguientes arrays y valores:

- a) El producto escalar (multiplicar los valores de los dos arrays que ocupan la misma posición y sumarlos)
- b) Sumar los dos arrays (array de 10 elementos cuyos elementos son la suma de los elementos que ocupan la misma posición en los dos arrays)
- c) Multiplicar los dos arrays (array de 10 elementos cuyos elementos son el producto de los elementos que ocupan la misma posición en los dos arrays)

(OperarArrays)

16. Se considera un array de 10 elementos con valores aleatorios entre -10 y 10. Se mostrará el array, y se obtendrá un array con los valores positivos y otro con los valores negativos. Se mostrarán los dos arrays anteriores. (ArraysPositivosNegativos)

Arrays de caracteres

17. Léase una frase y conviértase cada letra (mayúscula o minúscula) en su letra siguiente del abecedario siempre que sea posible (la letra 'z' no tiene siguiente, ni las vocales acentuadas ni la letra 'u' con diéresis). Se observa que la letra 'n' se convertirá en 'ñ', y la 'ñ' en 'o'. (FraseConvertir)

Arrays de cadenas

18. Léase una frase y obténgase lo siguiente:

- a) Las palabras más largas.
- b) Las palabras más cortas.

- c) Las palabras que tienen exactamente un número de caracteres que se introduce por consola.

(Palabras)

19. Rellenese un array de 10 cadenas con palabras aleatorias con letras mayúsculas entre 'A' y 'Z' (sin contar la 'Ñ', ni las vocales acentuadas, ni la 'Ü'). El número de caracteres de cada palabra es un número aleatorio entre 5 y 20. Se mostrarán dichas palabras en líneas distintas, y se indicará cuál es la primera y última en orden alfabético. (GenerarPalabras)
20. Se introducirán en líneas distintas el nombre y la edad de 10 personas separadas por espacio. A medida que se introduce cada línea se validará que tiene sólo dos campos, que el primero tiene un formato de nombre (palabra que comienza por mayúscula y el resto en minúsculas) y el segundo tiene formato de número natural entre 0 y 120. Una vez leídas correctamente las 10 líneas, se obtendrá la suma de las edades y total de veces que aparece la letra 'a' en todos los nombres. (NombresEdades)

Métodos de cadenas que devuelven un array

21. Lease una frase y muestrese en líneas distintas todas las subcadenas que están separadas por el carácter "|". (Split)

Métodos de clase que devuelven un array

22. Léase un entero grande y muéstrese su representación en complemento 2. (ComplementoDos)
23. Léanse dos enteros grandes y obténgase en una sola variable de tipo array su cociente y resto entero. (DivisionResto)

Ordenar arrays

24. Modifíquese el ejercicio "GenerarPalabras" del apartado "Arrays de cadenas" de forma que se muestren las palabras ordenadas. (GenerarPalabrasOrdenadas)
25. Genérese un array de 10 números reales entre 0 y 1, ordénese de mayor a menor y muéstrense en líneas distintas (RealesOrdenadosDescendentemente)
26. Genérese un array de 10 enteros entre 0 y 9, ambos inclusive. Ordénese de forma que aparezcan primeros los pares ascendentemente y luego los impares ascendentemente. (EnterosOrdenarParesImpares)

Recorrer una tabla

27. Cada vez que se ejecute una aplicación se rellenará una tabla de 5 x 5 de enteros con valores aleatorios entre 0 y 99, se mostrará por consola dicha tabla y se obtendrá:

- a) El producto de los pares.
 - b) El mayor y el menor número generado.
- (RecorrerTabla)

28. Resuélvese el ejercicio anterior leyendo previamente el número de filas y columnas (entre 3 y 20) que tiene la tabla. (RecorrerTablaGenerico)

29. Se considera una tabla de 10 x 10 de enteros cuyas celdas se inicializan a 0. Se generarán 100 celdas aleatorias a las cuales se les asigna el valor de 1. Se mostrará el total de celdas con 0, con 1 y el número de celdas aleatorias que se han repetido. (TablaCerosUnos)

Cálculo de arrays asociados a una tabla

30. Cada vez que se ejecute una aplicación se rellenará una tabla de 5 x 5 de reales con valores aleatorios entre 0.00 y 0.99, se mostrará por consola dicha tabla y se obtendrá:

- a) La suma de las celdas de cada fila.
- b) La suma de las celdas de cada columna.
- c) El o los números que más veces se repite.
- d) La suma de los elementos de la diagonal principal
- e) La suma de los elementos de la diagonal secundaria

(RecorreFilasColumnas)

31. Genere aleatoriamente las notas (con valores entre 0.0 y 10.0) obtenidas por los 10 alumnos de una clase de primer curso de DAM en cada uno de los módulos (“PRO”, “BD”, “SI”, “ED”, “FOL” y “XML”). Se obtendrá para cada alumno y para cada módulo la nota media con 1 decimal, el número de suspensos (notas inferiores a 5) y el número de sobresalientes (notas superiores a 8.4) (TablaNotas)

32. Cónsidérese una tabla de 3 x 8 de enteros con valores entre 0 y 9. En el programa se rellenará la tabla, se mostrará y se obtendrá la suma de cada una de las diagonales. (TablaDiagonales)

Métodos con tablas

33. Prográbase una sopa de letras. La sopa de letras será una tabla de 15 x 15 de caracteres. Las palabras, de como máximo 15 caracteres, se obtendrán de forma aleatoria de una tabla de cadenas por temas. Se elegirá un tema aleatorio y, para el tema elegido, se elegirán un máximo de 8 palabras como máximo.

(SopaDeLetras)

34. Simúlese el juego de los barquitos. Se generará el tablero con los barquitos y se intentarán hundir los barcos en un número máximo de intentos. (HundirLaFlota)

35. Hágase un formulario que permita jugar a las cuatro en raya en un tablero de 6 filas y 7 columnas. (CuatroEnRaya)

Expresiones de tipos de envoltorio

36. Léanse números enteros hasta introducir un 0 o un número negativo. Se obtendrá la media de los datos obtenidos. (IntegerMedia)

Conversión de cadenas a tipos envoltorio

37. Hágase un programa que lea dos enteros largos en binario y obtenga su suma. (SumarLongBinario)

Conversión de Integer y Long a base 2, 8 y 16

38. Hágase un programa que lea un número natural (se validará que es un número natural) y lo convierta a binario, octal y hexadecimal. (IntegerConvertir)
39. Léase una cadena en formato hexadecimal (se validará que es una cadena hexadecimal) y conviértase a binario. (LongConvertir)

Ordenar Arrays de tipo envoltorio

40. Hágase un programa que rellene un array con 10 enteros entre -999 y 999, ambos inclusive. Se mostrará el array separando los datos con un espacio. Se ordenará el array de forma que primero aparezcan los pares, luego los negativos y por último los positivos. (OrdenarArrayEnteros)
41. Hágase un programa que genere 10 reales con dos decimales entre -99.99 y 99.99. Se mostrará el array separando los datos con comas. Se ordenará de forma que primero se mostrarán los valores entre -10.00 y 10.00 en orden ascendente y luego el resto de los valores en orden descendente. (OrdenarArrayReales)

Recorrer un ArrayList

42. Se pretende calcular la nota obtenida por un alumno en una asignatura. Para ello se lee el número de pruebas que hizo el alumno. Para cada prueba se introduce la nota obtenida (de 0 a 10) y su peso (de 1 a 10). El cálculo de la nota se obtiene del siguiente modo:

$$(n_1 * p_1 + \dots + n_m * p_m) / (p_1 + \dots + p_m)$$

El profesor le muestra al alumno como fue evolucionando su nota final a medida que iba realizando las pruebas.

ENTRADA/SALIDA

Número de pruebas: **3**

Prueba 001

Nota: **7**

Peso: **3**

Prueba 002

Nota: **6**

Peso: **5**

Prueba 003

Nota: **3**

Peso: **2**

$7 * 6,4 * 5,7$

(la primer nota se obtiene teniendo sólo en cuenta la primera prueba; la segunda nota, la primera y segunda prueba; y la tercera nota, la primera, segunda y tercera prueba)

Ordenar un ArrayList

- 43.** Hágase un programa que lea enteros entre -9999 y 9999 hasta introducir el valor 0. Los datos se almacenarán en un ArrayList. El programa ordenará el ArrayList de forma que primero aparezcan los valores con menos de 3 dígitos ordenados de menor a mayor y luego el resto de valores ordenados de mayor a menor. (OrdenarArrayListEnteros)

Métodos con ArrayList

- 44.** Defínase un menú que permita insertar sin repeticiones, mostrar, eliminar valores enteros y vaciar dicho array dinámico. (MenuArrayListEnteros)

Pilas

- 45.** Hágase una aplicación que lea tres enteros, los almacene en una pila, recorra la pila y obtenga la suma de los elementos apilados. (PilaEnteros)

Conjuntos hash

- 46.** Hágase un programa que genere un boleto de la bonoloto (siete números distintos entre 1 y 49, ambos inclusive, y uno entre 0 y 9). Los siete números distintos se generan en un conjunto. (BonolotoConjunto)
- 47.** Hágase una aplicación que lee números enteros hasta introducir el 0, los almacena en un conjunto hash, y muestra la suma de los datos positivos y negativos introducidos. (ConjuntoHashEnteros)

Tabla hash

- 48.** Hágase una aplicación en la que almacenan en una “Hashtable” tres enteros que se leen con los índices “numero1”, “numero2” y “numero3”; se recorre la tabla y se obtiene la suma de sus elementos. (TablaHashEnteros)
- 49.** Considérese una tabla hash de clave única de tipo cadena (nombre de una persona), y con valores enteros (teléfono fijo o móvil). Se tendrá un menú que permita insertar, mostrar, vaciar, eliminar y modificar datos almacenados en dicha tabla hash. No se

pueden almacenar dos personas con el mismo nombre ni dos teléfonos iguales.
(MenuHashtableAgenda)