Arreglos

Los arreglos forman parte de nuestra vida cotidiana, los encontramos en los asientos de un camión, en las sillas de un salón, en las habitaciones de una casa, académicamente es un lugar en la memoria que guarda valores del mismo tipo al mismo tiempo, coloquialmente podría decir que son lugares donde guardamos cosas, por ejemplo una zapatera guardamos pares de zapatos.

En la computación los arreglos son variables compuestas de más variables es decir son una colección (agrupación o conjunto de ellas), en la figura 1 se muestra una colección de estaturas (alturas de personas), en donde debemos destacar 3 elementos:

1. El ***nombre*** de la colección, la cual se llama ***altura***
2. El ***índice*** que es un numero que asocia una posición (celda, casilla o lugar) dentro del arreglo también se le llega a llamar referencia, la cantidad de índices requeridos depende de la **DIMENSION** del arreglo, ***el índice no puede ser menor de 0, ni mayor a la cantidad de casillas contenidas menos 1.***
3. El ***valor*** de cada celda

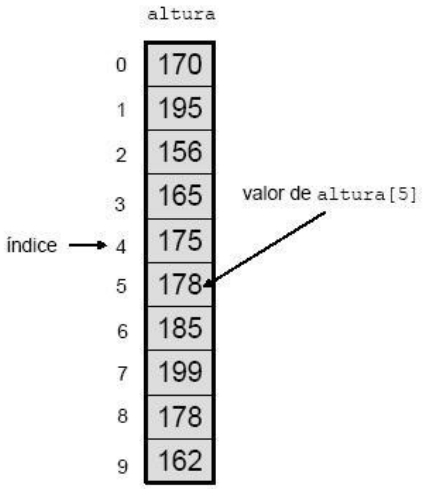


Figura 1. Definición grafica de un arreglo con valores de estaturas en centímetros de 10 personas (de la casilla 0 a la 9), solo se usa un índice por ser un arreglo ***UNIDIMENSIONAL***

Todos los arreglos en JAVA comienzan en la celda 0, aquí se presta para mal interpretar un juego de palabras por ejemplo que valor esta la posición 1 del arreglo, puede ser ambiguo, si no se aprende que posición=celda=referencia=índice, se podría pensar que es el 170 y no, el 170 esta en la posición cero.

Las agrupaciones de valores de un mundo real dependen de la abstracción que se haga del problema y en ocasiones se requieren de varias dimensiones (índices) para DIMENSIONAR el problema. Los arreglos se clasifican en dimensiones y lo que cambia es el prefijo, UNI, BI, TRI (prefijos griegos o latinos) y así sucesivamente los arreglos como en la de la figura 1 son arreglos del tipo unidimensional solo se requiere de un índice o dimensión para ubicar una celda y con ello su información.

En JAVA, declarar un arreglo se puede hacer de las siguientes formas:

1) Tipo_de_dato []nombre_arreglo;
2) Tipo_de_dato nombre_arreglo[];

3) Tipo_de_dato []nombre_arreglo={valor1,valor2,valor 3, …….};

En las 3 formas estamos definiendo un arreglo UNIDIMENSIONAL por lo tanto se requerirá de un índice, en las formas 1 y 2 los arreglos son objetos dinámicos, es decir, en principio no tienen una cantidad de celdas definidas. El complemento de estas definiciones es la construcción del objeto tipo arreglo y la cantidad de datos que va a coleccionar, como se muestra en el ejemplo 1.

Ejemplo 1	int edades[];
	edades=new int[200];
	int votos= new votos[300];
	String []apellidos={"gamez","alcantar","fuentes","mendoza","moreno"};

En los ejemplos anteriores se declaran los arreglos, uno para que contenga 200 casillas y el otro para 300, si mas adelante se requiere cambiar el tamaño, en JAVA, simplemente creamos otro arreglo con el operador ***new*** y listo, el arreglo anterior pasa a ser un bloque de memoria basura y JAVA tiene un recolector de memoria basura para tener el máximo de memoria disponible en nuestra computadora.

En la tercer forma se esta definiendo en principio un arreglo estático, es decir, ya tiene definido el numero de elemento y además se esta INICIALIZANDO en esos valores, este tipo de declaración es útil cuando tenemos pocos elementos pero si fueran 500 o 1000, inicializarlo así no seria algo practico. Es decir tenemos datos de 4 divisas se podría declarar de la siguiente forma ***float divisas[]={12.56,18,9,0.167,22.1};***

¿PERO POR QUE NECESITO ARREGLOS?

La solución de un problema no necesariamente requiere del manejo de arreglos, los arreglos como las variables, los ciclos y los otros elementos de la programación son herramientas como un martillo o una sierra eléctrica, el problema no es saber que existe ni como funciona sino saberlo usar, un ejemplo fácil, si no se sabe usar un martillo y quiero poner un clavo en la pared, ¿Qué sucederá? O si quiero cortar una madera con la sierra eléctrica y no la se usar, pues creo que es obvio que lo que pasara en que me lastimare, en el caso de los arreglos sucede algo parecido, si los uso mal el resultado será malo, los arreglos se usan tenemos una cantidad importante de datos o cuando determine que es la única herramienta que me ayudara a solucionar el problema.

¿PUEDO SOLUCIONAR CUALQUIER PROBLEMA SIN ARREGLOS?

Pues si, si declaras la cantidad de variables necesarias para cubrir la cantidad de información que requieras, por ejemplo en el caso de las divisas se podría solucionar con 4 variables una para cada divisa, pero ¿Que pasaría si tuvieras 200 divisas?, tendrías que crear 200 variables O_o.

Un arreglo se puede manipular de cualquier forma para introducir, buscar, modificar, o eliminar datos, estos procesos no son automáticos, al menos no en arreglos como éstos, existen otros tipos de datos compuestos como el ***vector*** o el ***arraylist*** que ya tienen definidos métodos específicos para manipularlos.

Loas arreglos en JAVA por ser objetos heredan propiedades como ***length***, el cual nos indica cuantas casillas tiene declarado el arreglo y aunque heredan otras propiedades a nivel básico no es necesario utilizarlas. Retomando las declaraciones el ejemplo 1, en el arreglo ***votos***, su manipulación podría ser como en el ejemplo 2.

Ejemplo 2 Guardar valores en el arreglo	votos[100]=45; // de forma directa, en la celda 100 guardamos un 45
	votos[ref]=34; // ref es una variable, parámetro o atributo de una clase, que no
	// puede valer menos de 0 ni mas de 299 , en la posición ***ref*** se
	// almacena un 34
	votos[3]= variScanner.nextInt(); // se puede asignar el valor de un proceso
	
	// Ejemplo de manejo de length con un ciclo para inicializar a CERO
	// todas las celdas del arreglo edades
	for (ref=0; ref<edades.length; ref++)
	edades[ref]=0;

Alto no siga leyendo: RESUELVA del EJERCICIO 1 al 9

Los arreglos unidimensionales son conocidos también como vectores, y son muy útiles para hacer cálculos sobre una cantidad enorme de datos, por ejemplo si tenemos el arreglo ***altura*** de la figura 1 y queremos saber cuanto se separan los datos entre si aplicaremos un concepto llamado Desviación estándar cuya formula se ve en la formula 1.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{N}}$$

Formula 1

La formula se interpreta como que tenemos ***n*** cantidad de datos y que se debe de calcular la diferencia al cuadrado de cada dato, menos el promedio de los datos. En otras palabras necesitamos calcular primero el promedio de los datos conocidos, para después por cada dato hacer una resta.

Por ejemplo si el arreglo tuviera los valores 5,8,2,8,4 (todos enteros) su promedio seria 27/5, ***OJO*** en JAVA esta división dará 5, y necesitamos que el resultado sea float o double, para obtener 5.4, con este valor necesitamos calcular: 5-5.4 elevarlos al cuadrado y sumarlo con 8-5.4, y así sucesivamente con todos los demás, hasta finalmente dividir la suma entre el numero de elementos que son y obtener su raíz. Un código que resuelve éste planteamiento se puede observar en el ejemplo 3:

Ejemplo 3. Código que resuelve el planteamiento de la desviación estándar.

```
public float  m_promedio(int p_datos[])
{  int v_indice;
   float v_suma=0,v_promedio;
```

```
  for( v_indice=0; v_indice<p_datos.length; v_indice++)
        v_suma+=p_datos[v_indice];
 v_promedio=v_suma/p_datos.length;
return v_promedio;
}
public double m_desvEstandar(int p_datos[])
{  int v_indice,v_resta;
  double v_division;
  Int v_sumatoria=0;

  for  (v_indice=0; v_indice<p_datos.length; v_indice++)
   {  v_resta= p_datos[v_indice]- m_promedio(p_datos);
      v_sumatoria+=Math.pow(v_resta,2);
   }
 v_division=  v_sumatoria/p_datos.length;
  return Math.sqrt(v_division);
}
```

Si tuviéramos miles de datos este código simplificaría los cálculos.

Alto no siga leyendo: Resuelva el Ejercicio 10 y 11

Los datos en muchos problemas vienen en pares como por ejemplo los datos de coordenadas cartesianas, en donde se tiene X y Y de los dos ejes de un sistema plano como la figura 1, cada punto coordenado tiene dos componentes.

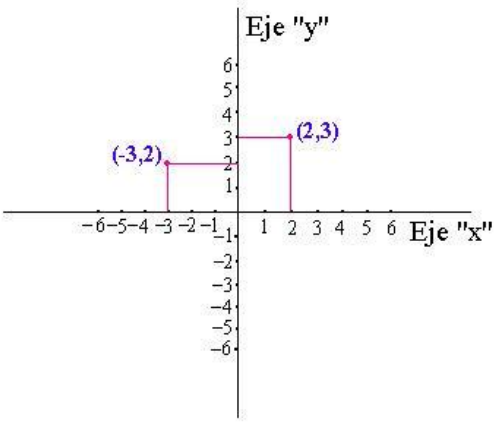


Figura 1 Sistema Cartesiano

Ahora bien si necesitáramos guardar un conjunto (colección) de estos puntos para posteriormente procesarlos, necesitaremos más de un arreglo, aquí aparece el concepto de ARREGLO PARALELO, que no es mas que dos o mas arreglos cuya información tiene una relación directa, esta relación se maneja por medio de los índices, así por ejemplo para almacenar los datos del problema planteado tendríamos que tener el arreglo coorX y coordY, en

donde coorX tendría los valores de las X y el otro el de las Y, si nuestros puntos coordenados fueran (-3,2),(2,3),(0,0),(5,-2) y (-1,-1.5), una forma de definir de forma estática los arreglos seria:

float v_coorX[]={-3,2,0,5,-1}, v_coorY[]={2,3,0,-2,-1.5};

El manejo de arreglos paralelos requiere de una mayor Abstracción o lógica de solución, la lógica a aprender es que cuando en el código haga uso de un índice de un arreglo, por lógica podre hacer uso de del valor del segundo arreglo.

Por ejemplo si uso coorX[3] para un proceso puede ser que requiera hacer uso de coorY[3], debido a que entre los 2 se define el punto coordenado 3.

Alto no siga leyendo: Resuelva el ejercicio 12 y 13

Otra idea de arreglos paralelos, retomando lo de las divisas, es que en el arreglo del ejemplo de las divisas solo guardamos el valor de la divisa mas no el nombre, de tal forma que no sabemos de forma natural a que divisa pertenece cada valor, aquí es donde un arreglo paralelo podría servir.

float divisas[]={12.56,18,9,0.167,22.1};
String nombDivisas[]={"Dólar","Euro","Yen","Libra Esterlina"};

De esta forma se relaciona de forma lógica, qué en la posición 0 se encuentra la información de del **dólar**, en la posición 1 se encuentra la información del euro y así sucesivamente. Cabe recalcar que un String es un objeto que se basa en un arreglo de caracteres para almacenar su información es por eso que se puede utilizar el método charAT para acceder a una letra en particular, en realidad estamos accediendo a una celda.

Resuelva el ejercicio 14

Ejercicio 1

Escribe el código en Java para declarar y crear los siguientes arreglos:

- Un arreglo de 100000 reales.
- Un arreglo de 1 valor booleano.
- Un arreglo de 450 enteros.
- Un arreglo de 150 String.

Ejercicio 2

Observa el siguiente código y obtén los valores almacenados en las celdas indicadas

```
int B[ ];
B = new int[5];
B[4]=1;
B[B[4]]=2;
B[B[B[4]]]=0;
B[B[B[B[4]]]]=3;
B[B[B[B[B[4]]]]]=4;
```

El valor de B[0] es ____
El valor de B[1] es ____
El valor de B[2] es ____
El valor de B[3] es ____
El valor de B[4] es ____

Ejercicio 3

Basados en tener un arreglo con la siguiente definición:
int datos[]={3,8,5,15,24,0,4,6}, a=2;
Que valor se tiene en las siguientes posiciones:
datos[5] : _____
datos[datos[5]] : _____
datos[datos[0]+datos[6]] : _____
valor de la celda posición 2+ valor de la celda posición 8 :_____
datos[a*3]: _____
datos[a/3]: _____

Ejercicio 4.

Basados en el arreglo datos del Ejercicio 3: Cree un método de una clase que reciba como parámetro el arreglo y regrese cuantos impares están dentro del arreglo, utilice el length.

Ejercicio 5

Haga una prueba de escritorio, ya sea mental o en su libreta del código siguiente y obtenga el valor de que se pide al final del ejercicio

```
double A[ ];
A = new double[3];
int x;
for (int i=0; i<A.length; i++)
    A[i] = i*3;
x=A[0]+A[1]+A[2];
```

El valor de x es ____
El valor de A[0] es ____
El valor de A[1] es ____
El valor de A[2] es ____

Ejercicio 6

```
int C[ ];
C = new int[4];
int y;
for (int i=C.length-1; i>=0; i--)
    C[i] = i/2;
y=C[0]+C[1]+C[2]+C[3];
```

El valor de y es ____
El valor de C[0] es ____
El valor de C[1] es ____
El valor de C[2] es ____
El valor de C[3] es ____

Ejercicio 7.

Haga un programa que pida el numero de casillas que tendrá un arreglo y luego lo llene con valores booleanos alternados, ejemplo si son 4 casillas las que se generan se almacenará true,false,true,false, desplegándolas al final en la consola.

Ejercicio 8.

Se tiene un arreglo de Cadenas de caracteres en donde guardamos apellidos, haga un método que reciba como parámetro el arreglo y un apellido y regrese cuantas veces esta el apellido en el arreglo.

Ejercicio 9.

El sabueso; Hay que declarar un arreglo con los siguientes datos { 2,9,5,8,7,1,6,3,-1,3}, este arreglo será revisado de la siguiente forma, se empieza en la celda 0 y de la celda se toma el valor para la siguiente referencia ósea el 2, ahora se revisa la celda 2 y se obtiene el 5, hay que hacer este proceso de forma continua hasta encontrar el -1 que es el fin de las pistas, ahí termina el sabueso de seguir pistas, se debe

de ir contando los rastros. Cuando este su código completo compruébelo con este otro arreglo. {15,10,-1,1,5,11,9,8,2,13,5,14,0,2,7,3}

Ejercicio 10.

Reduzca el código utilizando la menor cantidad de variables, este es un ejercicio de abstracción.

Ejercicio 11.

La cruz Roja Mexicana esta haciendo una colecta en donde al final del día quiere saber cuantas mujeres donaron, cuantos hombres y el monto total, haga un programa que resuelva esta situación, en donde el único atributo de la clase es un arreglo, no puede declarar mas atributos y si puede declarar una variable/atributo tipo Scanner para la lectura de los datos. El ejercicio consiste en manipular en un arreglo todas las variables necesarias que ayudan a resolver el problema y aunque cree varios métodos, éstos no declaren variables si no que utilicen el atributo de la clase (que es un arreglo).

Ejercicio 12

Basados en los arreglos paralelos coorX y coorY, de la explicación de arreglos paralelos haga 2 métodos:

a) El primero que busque un punto coordenado y regrese en que posición está
b) El segundo, conocida la referencia obtenga las distancia que hay entre el origen 0,0 y los valores dados encontrados en la referencia. Para calcular la distancia se hace obteniendo la raíz cuadrada de la suma de los cuadrados de los valores , por ejemplo si fuera la referencia 3, distancia= Raíz (5 al cuadrado + -2 al cuadrado)

Ejercicio 13

Basados en los arreglos paralelos coorX y coorY y en la formula de la distancia entre 2 puntos haga un método que calcule y guarde en un arreglo mas la distancia de cada par de puntos.
Distancia entre 2 puntos = Raíz cuadrada [(x2-x1) al cuadrado +(y2-y1)al cuadrado]

Ejercicio 14.

Utilizando este concepto de arreglos paralelos, haga un programa que tenga un menú con 4 opciones:

1. agregar la hora de entrada (H:M:S una en cada arreglo)
2. Listar las horas,
3. Borrar un renglón, recorriendo a todos los reglones posteriores.
4. Salir del programa

Ejercicios Extras en EQUIPO

1. Escriba un método que reciba como parámetro un arreglo de datos enteros y 2 números enteros (anterior y nuevo), y lo que debe de hacer es cambiar todos los números del arreglo que sean iguales al anterior por el nuevo.
2. Escriba un método que reciba un arreglo de datos String y de forma interna invierta el arreglo, quedando modificado, aquí interviene el concepto de parámetro por referencia. Ejemplo si recibe {“hola”, ”mundo”, ”de”, “POO”}, lo debe de cambiar a que quede {“POO”, ”de”, ”mundo”, ”hola”}
3. Escriba un método que reciba como parámetro un arreglo de datos enteros, un numero entero y una cadena que solo puede ser “<<” o “>>”, el método hará una ROTACIÓN de posiciones algo parecido a lo que hace el operador >> o <<, dependiendo de la cadena, la rotación será hacia la celda 0 o hacia la celda N del arreglo, y la cantidad de rotaciones que hará será el número pasado como parámetro.
   Ejemplo si el arreglo contuviera {4,8,9,2,3,5} el entero fuera 2, y la cadena >>, el arreglo quedaría así {3,5,4,8,9,2}.
4. Haga un programa que por medio de un método determine si dos arreglos del mismo tipo son iguales en todo.
5. Del Ejercicio 9, haga un programa que utilice el código que se genero, el código que genero debe de mejorarse para que nunca se rebase el tamaño del arreglo y por otra parte se dé cuenta de que ya paso por una pista y en ese instante se detenga., es decir, para casos como los siguientes {3,7,1,9,5} y {5,4,2,3,0,1}, en el primer caso al llegar a la posición 3 ésta nos mandaría a la posición 9 que no existe, en el segundo caso cuando nos mandan a la posición 4 ésta nos regresa a la 0 y por la 0 ya pasamos (ojo la condición de repetición no es pasar por cero, sino por cualquiera que ya hayamos pasado).
6. Haga un programa que maneje arreglos paralelos como los asientos de un camión de pasajeros, y permita comprar un boleto o cancelarlo (liberando el asiento), los datos del pasajero son 3 su nombre, el asiento que compro y el precio que pago, el programa debe preguntar al inicio cuantos asientos tiene el camión y posterior un menú de opciones Comprar 1 boleto, mostrar estatus de los asientos, mostrar el dinero recabado hasta ese instante, cancelar boleto y Salir.
7. Haga un programa qué, dado un conjunto de puntos COORDENADOS (X y Y), determine la suma de las áreas rectangulares que se forma entre 2 puntos contiguos. Ejemplo si tuviéramos los puntos (1,1)(3,3)(5,5)(4,1)(6,2) como se muestra en la figura 2.

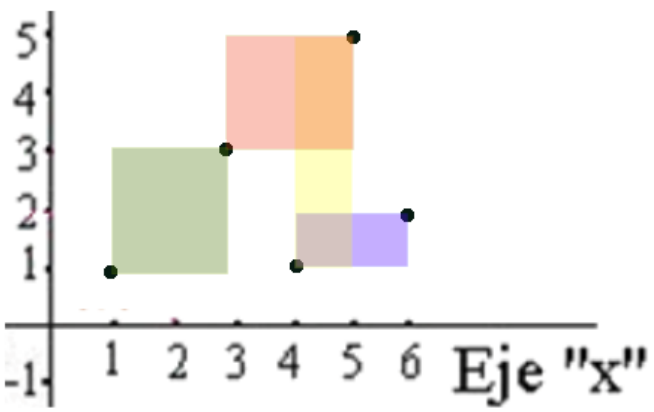


Figura 2. Puntos coordenados que generan Rectángulos

El resultado seria: 14 (4+4+4+2)

Arreglos de Objetos.

Arreglos Bidimensionales

Ejercicio 1. Haga un programa que construya una matriz y después pida cuantas veces se rotara la matriz, y la gire o le haga una rotación, rotar se refiere a desplazar el contenido de una celda hacia otra celda una posición, guíese con el ejemplo:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Girada 2 veces queda así =>

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

1er. Giro

16	15	14	13
12	11	10	9
8	7	6	5
4	3	2	1

2º Giro

Ejercicio 2. Haga un programa que cargue una matriz con números del 1 al 4, le pida al usuario la posición de inicio (renglón, columna), el objetivo es tomar a los 4 números como una guía a seguir, el 1 significa moverse a la derecha, el 2 hacia abajo, el 3 hacia la izquierda, y el 4 hacia arriba (los arreglos no tiene esas direcciones, sin embargo como ejemplo practico se considerará así). Partiendo de la posición dada por el usuario se moverá un robot imaginario hacia la celda que indica hasta llegar a un punto fuera de la matriz contando los pasos, por ejemplo si empezamos en cualquier posición de la matriz siguiente, el programa termina por encontrar la salida, pero si empezamos en 2,0 ó 3,0 se cicla, el programa debe evitar que se cicle terminando y dando el mensaje pertinente, si empezamos en la celda 2,3 en la figura se muestra el recorrido, en 9 pasos.

4	3	2	3
1	1	2	4
2	2	3	4
4	1	2	4

		4	3
		5	2
	7	6	1
	8	9	