

Arreglos

¿Porque usar Arreglos?

- Se tienen las calificaciones de 100 alumnos, ¿como sacar cuantos alumnos tienen una calificación superior al promedio?
- Doble lectura:
 - Un ciclo para leer y acumular las calificaciones, y posteriormente calcular el promedio.
 - Un ciclo para comparar cada calificación con el promedio y sacar el valor superior.
- Definir 100 variables, una por cada alumno.

Doble Lectura

```
suma=0;
for (int i=1;i<=100;i++)
{ calf=Integer.parseInt(lector.readLine());
suma+=calf;}
promedio=suma/100;c=0;
for (int i=1;i<=100;i++)
{calf=lector.readLine();
  if (calf>promedio) c++;}
System.out.println("Mayores al promedio="+c);
```

Muchas Variables

Leer $c_1, c_2, c_3, c_4, c_5, \dots, c_{100}$

$suma = suma + c_1 + c_2 + c_3 + c_4 + c_5 + \dots + c_{100}$

$promedio = suma / 100$; $cont = 0$;

if ($c_1 > promedio$) $cont++$;

if ($c_2 > promedio$) $cont++$;

if ($c_3 > promedio$) $cont++$;

...

$System.out.println(cont)$;

Arreglos

- ¿Como decirle al computador que $c_1, c_2, c_3 \dots c_{100}$ son todas calificaciones?
- ¿Cómo acceder a todas con un ciclo for?
- Se hace necesario especificar “grupos” de variables
 - Todas del mismo tipo.
 - Todas con un nombre parecido.
 - Definidas con una secuencia especifica.

Definición de Arreglo

Es una colección de elementos y es:

- **Finita:** todo arreglo tiene un límite
- **Homogénea:** todos los elementos son del mismo tipo de dato.
- **Ordenada:** se puede determinar por el índice la posición de un elemento dentro del arreglo. El índice esta ordenado.

Declaración de Arreglos

`<tipo> <nombre> [] ;`



Cualquier
tipo válido

Cualquier
nombre válido

Corchetes
obligatorios

```
int ejemplo[];  
int a[]={10,20,30};  
int b[]=new int[10];
```

Referencia a elementos del arreglo

Se utiliza:

- el nombre del arreglo
- el índice del elemento

$x[10]=100;$

$var=x[5];$

$X[i+1]=x[i]+x[i-1];$

Observaciones

- El tipo del índice puede ser cualquier tipo ordinal (carácter, entero, enumerado).
- El tipo de los componentes o elementos puede ser cualquier tipo de datos u objeto.
- Se utilizan los corchetes [] para indicar el índice de un arreglo. Entre los [] se debe escribir un valor ordinal (puede ser una variable, una constante o una expresión que de cómo resultado un valor ordinal).

Dimensión de un arreglo

- Es la asignación y organización de los espacios de memoria asignados a un arreglo.
- La cantidad de índices necesarios para referenciar a un elemento o casilla del arreglo definen la dimensión del arreglo (o la cantidad de pares de corchetes en la declaración).

`int x[][];` Arreglo Bidimensional

`A[1][2][3]=100;` Arreglo Tridimensional

`B[i]=0;` Arreglo Unidimensional

Operaciones con Arreglos

- Recorrido
 - Lectura/Escritura
 - Asignación
 - Calculos
- Actualización
 - Inserción
 - Eliminación
 - Modificación
- Ordenación
- Búsqueda

Operación de Recorrido

- Recorrido completo del arreglo:
 - Se requiere un ciclo o bucle por cada dimensión del arreglo.
 - Un índice (variable) por cada dimensión.
 - El rango del índice va desde el limite inferior hasta el limite superior de la dimensión.

- Recorrido parcial del arreglo:
 - Se requiere un ciclo o bucle por cada dimensión en la que el índice varia de un valor inicial a un valor final, dentro del arreglo.
 - El rango del índice puede variar entre los valores definidos entre el limite inferior hasta el limite superior de la dimensión.
 - Los índices que permanecen constantes solo se utilizan para hacer referencia a una casilla y no cambia su valor.

Operaciones de Recorrido

- **Lectura:** consiste en leer y asignar un valor a cada uno de sus componentes.
- **Escritura:** consiste en mostrar o imprimir cada uno de sus componentes
- **Asignación:** consiste en asignar un valor específico o inicial, a todas o algunas casillas del arreglo.
- **Calculos:** consiste en realizar algún calculo con los componentes del arreglo (suma, promedio, máximo, menor, etc.)

Ejercicio

- Realizar un programa que lea un arreglo unidimensional que representa las ventas anuales de una empresa y calcular lo siguiente:
 - El promedio anual de ventas
 - ¿Cuáles son los meses que tuvieron ventas menores al promedio?
 - ¿Cuál es el porcentaje de ventas mayores al promedio?

Operaciones de Actualización en Arreglos Desordenados

Inserción:

1. Verificar que existan casillas disponibles ($n < \text{max}$)
2. Si se cumple:
 - Incrementar el total de elementos del arreglo ($n++$)
 - Asignar a la posición n el nuevo elemento ($\text{arreglo}[n] = \text{nuevoValor}$)
3. Si no se cumple: “No hay espacio en el Arreglo”

Operaciones de Actualización en Arreglos Desordenados

Eliminación:

1. Verificar que existan elementos en el arreglo ($n \geq \text{LimiteInferior}$, en java $n \geq 0$)
2. Si se cumple:
 - Buscar el elemento a eliminar dentro del arreglo, encontrar posición (POS).
 - Si existe:
 - Recorrer todos los elementos que están a la derecha del elemento a eliminar una posición a la izquierda (hacer desde $i = \text{POS}$ hasta $n-1$: `arreglo[i]=arreglo[i+1]`)
 - Decrementar n ($n--$)
 - No existe: “Elemento NO EXISTE”
3. Si no se cumple: “Arreglo vacío”

Operaciones de Actualización en Arreglos Desordenados

Modificación:

1. Verificar que existan elementos en el arreglo ($n \geq \text{LimiteInferior}$, en java $n \geq 0$)
2. Si se cumple:
 - Buscar el elemento a modificar dentro del arreglo, encontrar posición (POS).
 - Si existe:
 - Actualizar la casilla donde se encontró el elemento a modificar por el nuevo elemento ($\text{arreglo}[\text{POS}] = \text{nuevoElemento}$)
 - No existe: “Elemento NO EXISTE”
3. Si no se cumple: “Arreglo vacío”

Operaciones de Actualización en Arreglos Ordenados

Debe evitarse alterar el orden existente en el arreglo.

Inserción:

1. Verificar que existan casillas disponibles ($n < \text{max}$).
2. Si se cumple:
 - Buscar la posición que le correspondería al nuevo elemento para no alterar el orden existente en el arreglo (encontrar POS).
 - Incrementar n ($n++$)
 - Recorrer todos los elementos desde la N -ésima posición hasta la posición detectada ($\text{POS}+1$), un lugar a la derecha (ciclo decreciente: $\text{arreglo}[i]=\text{arreglo}[i-1]$)
 - Asignar el nuevo elemento en la posición encontrada ($\text{arreglo}[\text{POS}]=\text{nuevoElemento}$).

Nota: los pasos del recorrimiento no se llevan a cabo cuando el valor a insertar es mayor que el ultimo elemento del arreglo.

3. Si no se cumple: “No hay espacio en el Arreglo”.

Operaciones de Actualización en Arreglos Ordenados

Eliminación:

1. Verificar que existan elementos en el arreglo ($n \geq \text{LimiteInferior}$, en java $n \geq 0$)
2. Si se cumple:
 - Buscar el elemento a eliminar dentro del arreglo, encontrar posición (POS).
 - Si ($\text{arreglo}[\text{POS}] = \text{elementoBuscado}$) entonces:
 - Recorrer todos los elementos que están a la derecha del elemento a eliminar una posición a la izquierda (hacer desde $i = \text{POS}$ hasta $n-1$: $\text{arreglo}[i] = \text{arreglo}[i+1]$)
 - Decrementar n ($n--$)
 - En caso contrario: “Elemento NO EXISTE”
3. Si no se cumple: “Arreglo vacío”

NOTA: La operación de búsqueda esta optimizada, ya que se aprovecha el orden existente en el arreglo de tal forma que la búsqueda se realizará mientras que $\text{arreglo}[i] < \text{elementoBuscado}$.

Operaciones de Actualización en Arreglos Ordenados

Modificación:

- ❑ Se procede de manera similar a la eliminación
- ❑ Al modificar un valor X por un valor Y , debe verificarse que el orden del arreglo no se altere.
- ❑ Si esto llega a suceder, entonces podría rechazarse la operación o reordenar el arreglo.
- ❑ Como el reordenamiento es una operación costosa, lo mas conveniente es realizar una operación de eliminación del elemento X y una inserción del elemento Y .

Operaciones de búsqueda

- **Búsqueda Secuencial:** consiste en revisar elemento por elemento hasta encontrar el dato buscado, o hasta llegar al final de la lista de datos disponibles. Generalmente se utiliza cuando el arreglo esta desordenado
- **Búsqueda Binaria:** consiste en dividir el intervalo de búsqueda en 2 partes, comparando el elemento buscado con el central. En caso de no ser iguales se redefinen los extremos del intervalo (según el elemento central sea mayor o menor que el buscado) disminuyendo el espacio de búsqueda. El proceso concluye cuando el elemento es encontrado, o bien cuando el intervalo de búsqueda se anula. Este método se aplica solo en arreglos ordenados.

Búsqueda Secuencial

```
public boolean existe(int v[],int x,int n)
{ int i=0;
  while ((i<=n) && (v[i]!=x))
    i++;
  if (i>n) return(false);
  else return(true);
}
```

Donde v es el arreglo de n casillas, y x el elemento buscado

Búsqueda Binaria

```
public boolean existe(int v[],int x,int n)
{ int izq=0; int der=n;int cen=(izq+der)/2;
  while ((izq<=der) && (x!=v[cen])) {
    if (x>v[cen]) izq=cen+1;
    else der=cen-1;
    cen=(izq+der)/2;}
  if (izq>der) return(false);
  else return(true); //cen es la posición del elemento
}
```

Donde v es el arreglo de n casillas, y x el elemento buscado

Operaciones de Ordenamiento

Intercambio Directo (burbuja)

- La idea básica de este algoritmo consiste en comparar pares de elementos adyacentes e intercambiarlos entre sí hasta que todos se encuentran ordenados.
- Se realizan $(n-1)$ pasadas, transportando en cada una de las mismas el menor o mayor elemento (según sea el caso) a su posición ideal. Al final de las $(n-1)$ pasadas los elementos del arreglo estarán ordenados.
- El método de intercambio directo puede trabajar de dos maneras diferentes:
 - Llevando los elementos más pequeños hacia la parte izquierda del arreglo.
 - Llevando los elementos más grandes hacia la parte derecha del mismo.

Ejemplo Burbuja

Supóngase que desean ordenar las siguientes claves del arreglo A, transportando en cada pasada el menor elemento hacia la parte izquierda del arreglo.

A: 15 67 08 16 44 27 12 35

Las comparaciones que se realizan son las siguientes:

PRIMERA PASADA

$a[6] > a[7]$ ($12 > 35$) no hay intercambio

$a[5] > a[6]$ ($27 > 12$) si hay intercambio

$a[4] > a[5]$ ($44 > 12$) si hay intercambio

$a[3] > a[4]$ ($16 > 12$) si hay intercambio

$a[2] > a[3]$ ($08 > 12$) no hay intercambio

$a[1] > a[2]$ ($67 > 08$) si hay intercambio

$a[0] > a[1]$ ($15 > 08$) si hay intercambio

A: **08** 15 67 12 16 44 27 35

Obsérvese que el elemento 08 fue situado en la parte izquierda del arreglo.

Ejemplo Burbuja

SEGUNDA PASADA :

$a[6] > a[7]$ ($27 > 35$) no hay intercambio

$a[5] > a[6]$ ($44 > 27$) si hay intercambio

$a[4] > a[5]$ ($16 > 27$) no hay intercambio

$a[3] > a[4]$ ($12 > 16$) no hay intercambio

$a[2] > a[3]$ ($67 > 12$) si hay intercambio

$a[1] > a[2]$ ($15 > 12$) si hay intercambio

A: 08 12 15 67 16 27 44 35

y el segundo elemento más pequeño del arreglo, en este caso 12, fue situado en la segunda posición.

A continuación se muestra el resultado de las siguientes pasadas:

3era. Pasada : 08 12 15 16 67 27 35 44

4ta. Pasada : 08 12 15 16 27 67 35 44

5ta. pasada : 08 12 15 16 27 35 67 44

6ta. Pasada : 08 12 15 16 27 35 44 67

7ma. Pasada : 08 12 15 16 27 35 44 67

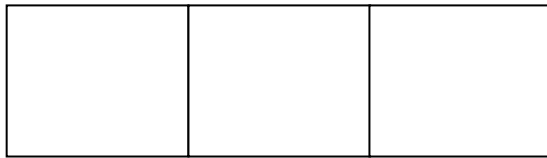
Algoritmo Burbuja

```
public void burbuja(){
    int tmp;boolean intercambio=true;
    for (int i=1;(i<=n)&&(intercambio);i++)
    {intercambio=false;
        for (int j=n;j>=i;j--)
            if (v[j]<v[j-1])
                { tmp = v[j];
                  v[j]=v[j-1];
                  v[j-1]=tmp;
                  intercambio=true;
                }
    }
}
```

Arreglos multidimensionales

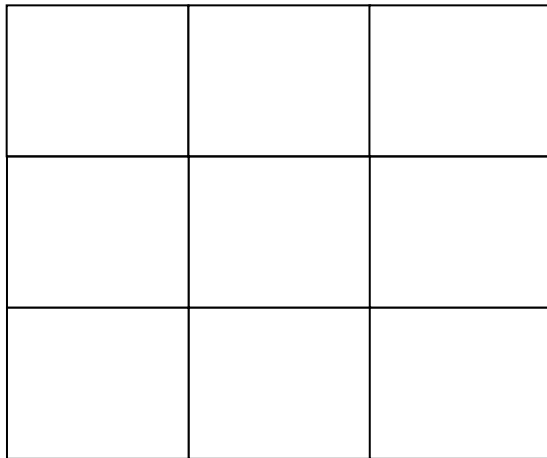
- Un arreglo de tamaño n puede verse como una matriz $1 \times n$.
- ¿Cómo definir matrices de $m \times n$?
- ¿Cómo definir matrices de $r \times m \times n$?
-

Arreglos multidimensionales



1x3

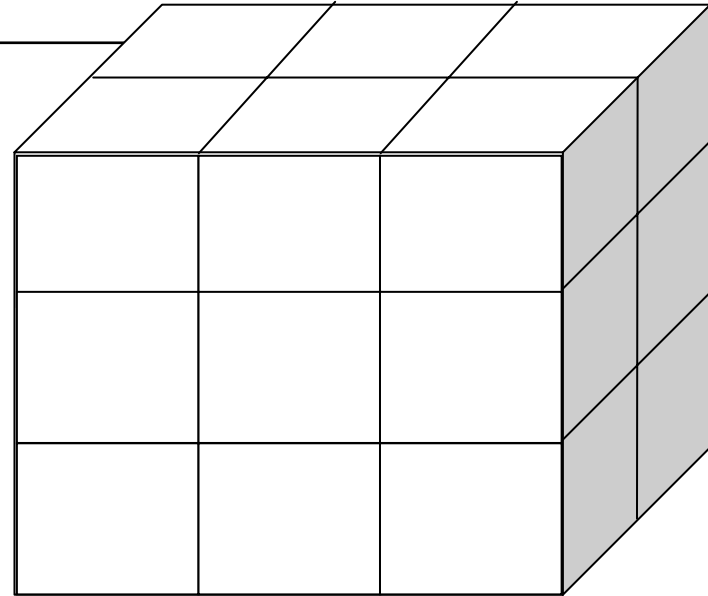
Arreglo de tres variables



3x3

Arreglo de 9 variables

¿O arreglo de 3 arreglos de 3 variables?



3x3x2

Arreglo de 18 variables

¿O arreglo de 2 arreglos de 3
arreglos de 3 variables?



Arreglos multidimensionales

- Las dimensiones en los arreglos se agregan con mas pares de corchetes.
- Nos limitaremos a ejemplos de matrices bidimensionales

Arreglos multidimensionales

A[fila][columna]

Fila del arreglo

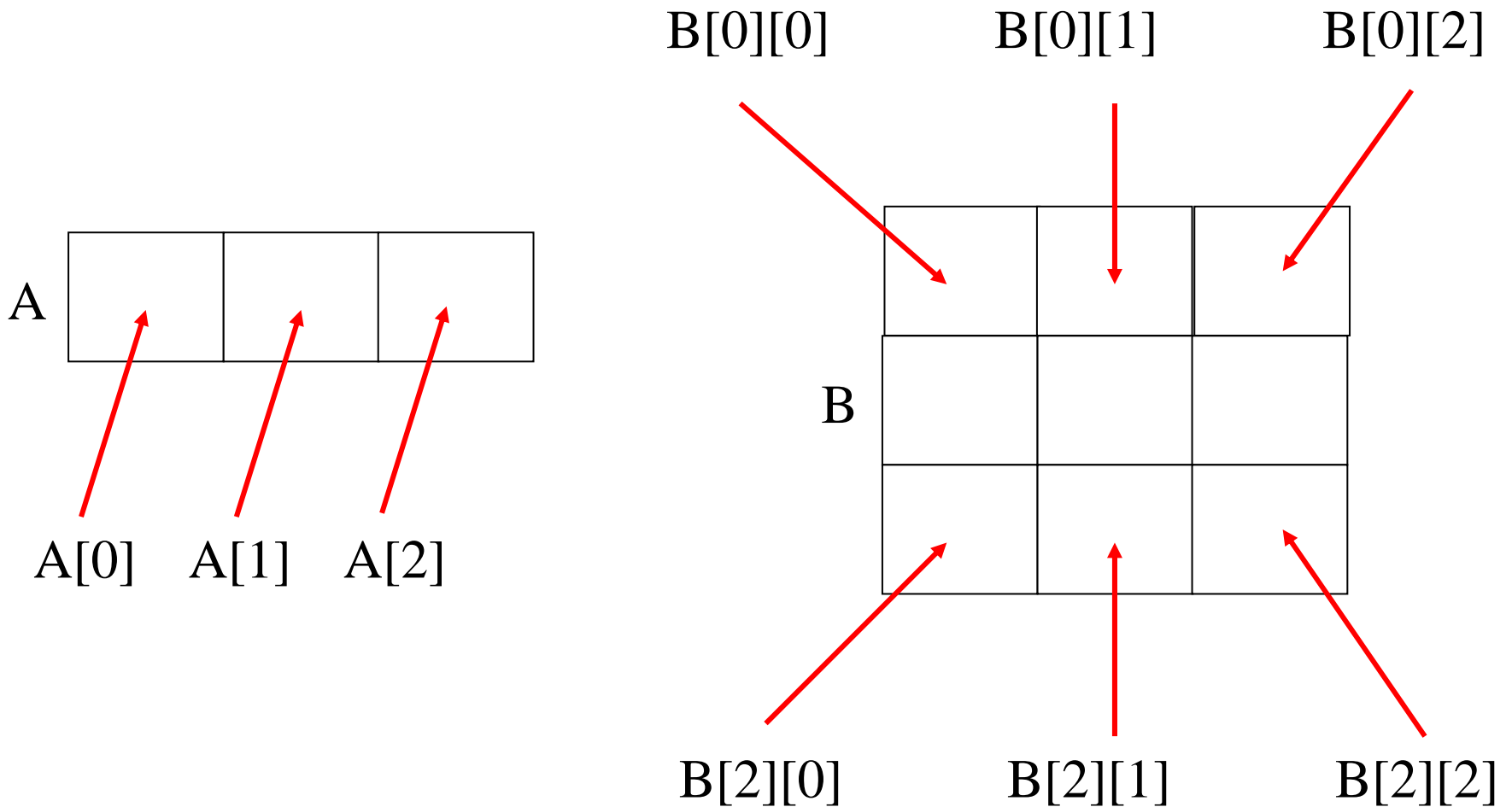


Columna del arreglo



B[fila][columna][prof]

Arreglos multidimensionales



Operaciones con Arreglos multidimensionales

- Se pueden realizar las mismas operaciones que en los arreglos unidimensionales.
- Las operaciones dependerán de la aplicación o del problema a resolver.
- Una operación se puede realizar de diferentes maneras según el número de dimensiones del arreglo (por renglones, por columnas, etc.)

Ejercicio

Una empresa lleva un registro del total producido mensualmente por departamento. La empresa consta de 5 departamentos y la información se ha registrado a lo largo de los últimos 4 años. Para almacenar la información se deberá utilizar un arreglo de 3 dimensiones ($5 \times 12 \times 4 = 240$ casillas). Escriba el código necesario para calcular:

- a) El total mensual de cada departamento durante el segundo año.
- b) El total de producción durante el primer año.
- c) El total de producción del departamento 3 a lo largo del último año.