ARRAYS (ARREGLOS)

2.1 INTRODUCCIÓN

- ✓ Los arreglos son **estructuras de datos** que consisten en elementos de información del mismo tipo relacionados entre sí. Los arreglos son entidades "**estáticas**" en cuanto a que su tamaño no cambia una vez que han sido creadas.
- ✓ Un arreglo es un grupo de posiciones de memoria contiguas. Todas las cuales tienen el mismo nombre y el mismo tipo.
- ✓ Los arrays pueden ser unidimensionales (**vectores**) ó bidimensionales (**matrices**)

Ejemplo:

VECTOR

12	14	17	8	19	13	7	9	6	92

MATRIZ

3	5	7	24
4	6	10	4
3	5	7	8

2.2 VECTORES

Cómo algunos ejemplos de vectores podríamos tener:

Vector de Enteros

12	14	17	8	19	13	7	9	6	92

Vector de Reales

1.3	0.05	4.0	6.7	1.0	4.7	9.002

Vector de Caracteres

'a'	'8'	'?'	'j'	'1'	'-'	'%'

Vector de Cadenas

"Bolivia"	"Argentina"	"Perú"	"Uruguay"	"Brasil"

Un vector debe tener un nombre (**sin espacios**) Por ejemplo

vec = | 12 | 14 | 17 | 8 | 19 | 13 | 7 | 9 | 6 | 92 |

Nombre del vector — Elementos del vector **vec**

Cada elemento de un vector tiene una posición, la misma que empieza en **cero**

vec =

12	14	17	8	19	13	7	9	6	92

POSICIONES→ **vec[0] vec[1] vec[2] vec[3] vec[4] vec[5] vec[6] vec[7] vec[8] vec[9]**

Entonces podemos ver que cada elemento de un vector tiene una posición y un dato
Por ejemplo:

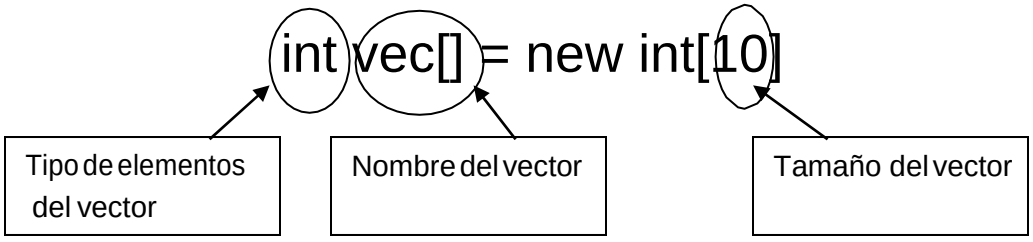
vec[0] tiene el dato 12
vec[3] tiene el dato 8
vec[8] tiene el dato 6
……..

Cada elemento del vector puede ser manejado como cualquier variable. Por ejemplo:

```
int A = vec[0] + vec[8];        // A = 12 + 6 =  18
int B = 2 + vec[3];             // B = 2 + 8 = 10
vec[0] = A + B;                 // vec[0] = 18 + 10 = 28
```

2.3 Declaración de vectores en JAVA

Los arreglos ocupan espacio en la memoria. El programador especifica el **tipo** de los elementos y usa el operador **new** para asignar espacio de almacenamiento al número de elementos requerido para el arreglo. Entonces para declarar al vector **vec** de los ejemplos anteriores sería:



En JAVA una vez creado un vector con datos numéricos los datos del vector por defecto se inicializan en **cero**

vec =

0	0	0	0	0	0	0	0	0	0

Nota. Un error muy común al programar con vectores es manejar posiciones que no existen en el vector. Por ejemplo en el vector **vec** no se podría utilizar el elemento vec[10] ó vec[11], porque no existen las posiciones 10 y 11. Ese error mostraría el siguiente mensaje **`java.lang.ArrayIndexOutOfBoundsException`**

2.4 Ejercicios con vectores

Realizar un programa para visualizar los datos de un vector

```
class vectores1
{
        public static void main(String args[])
        {
                int vec[]={2,3,4,5,6,7};
                for(int i=0;i<=5;i++)
                        System.out.println(vec[i]);
        }
}
```

Se puede dar valores al vector al momento de declarar el vector

Insertar los primeros 10 números naturales en un vector y posteriormente visualizar los datos del vector

```
class vectores2
{
	public static void main(String args[])
	{
		int A[]=new int[10]; //declaración del vector A de tamaño 10 de tipo int
		int con=0;
		while(con<10)
		{
			A[con]=con+1; //asignación de valores al vector A
			con++;
		}
		con=0;
		while(con<10)
		{
			System.out.println(A[con]); //muestra en pantalla los valores del vector A
			con++;
		}
	}
}
```

Programa para insertar por teclado 10 datos en un vector y posteriormente visualizar los datos.

```
 import java.util.*;
class vectores3
{
   public static void main(String args[])
   {
		Scanner en=new Scanner(System.in);
		int B[]=new int[100];
		int con=0;
		while(con<10)
		{
			B[con]=en.nextInt(); // asignación de valores mediante teclado
			con++;
		}
		con=0;
		System.out.println("los datos del vector son:");
		while(con<10)
		{
			System.out.println(B[con]);
			con++;
		}
   }
}
```

Programa para insertar las notas de 5 alumnos en un vector **notas** y posteriormente calcula el promedio

```
import java.util.*;
class vectores4 {
   public static void main(String args[])
   {
      Scanner en=new Scanner(System.in);
      float notas[]=new float[5];
      int con=0;
      float promedio,sum=0;
      while(con<5)
      {
         notas[con]=en.nextFloat();
         sum=sum+notas[con];
         con++;
      }
      promedio=sum/5;
      System.out.println(promedio);
   }
}
```

Programa para insertar 5 números en un vector A, copia en un vector B todos los datos pares del vector A, y posteriormente visualice los datos del vector B.

```
import java.util.Scanner;
class vectores5
{
   public static void main(String args[])
   {
      Scanner en=new Scanner(System.in);
      int A[]=new int[100];
      int B[]=new int[100];
      int i,con=0;
      for(i=0;i<5;i++)
         A[i]=en.nextInt();

      for(i=0;i<5;i++)
      {
         if(A[i]%2==0)
         {
            B[con]=A[i];
            con++;
         }
      }
      for(i=0;i<con;i++)
         System.out.println(B[i]);
   }
}
```

2.5 MATRICES

Una matriz es un array bidimensional (2 dimensiones, filas y columnas)

Cómo algunos ejemplos de matrices podríamos tener:

Matriz de Enteros →

3	7	2	6
4	6	4	8
2	4	6	8

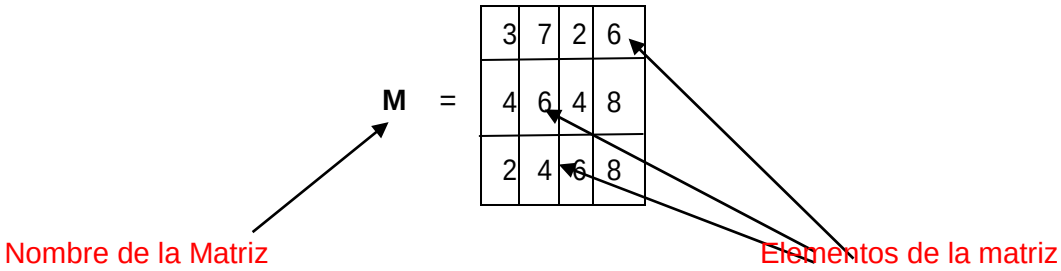
Matriz de Reales →

3.0	7.4	2.5	6.0
0.6	0.6	4.9	8.4
2.6	0.4	6.0	0.8

Matriz de Caracteres→

‘a’	‘B’	‘3’	‘0’
‘=’	‘/’	‘8’	‘y’
‘h’	‘.’	‘{‘	‘i’

Una matriz debe tener un nombre (**sin espacios**) Por ejemplo



Cada elemento de una matriz tiene una posición dado por la fila y columna, las mismas que empieza en **cero**

Sea la Matriz **M**

Posición filas	0	1	2	3
0	3	7	2	6
1	4	6	4	8
2	2	4	6	8

Posición columnas →

Matriz de tamaño 3 X 4 (3 filas y 4 columnas)

Entonces podemos ver que cada elemento de una matriz tiene una posición (dado por la fila y columna) y un dato

Por ejemplo:

M[0][1] tiene el dato 7
M[3][2] error porque no existe la fila 3
M[2][0] tiene el dato 2
M[2][3] tiene el dato 8

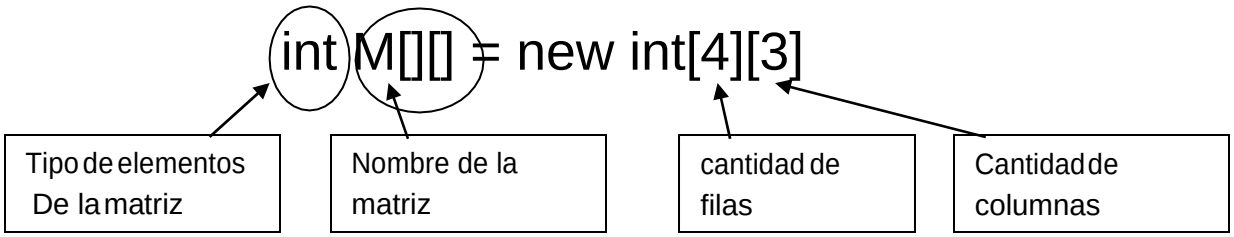
……..

Cada elemento del vector puede ser manejado como cualquier variable. Por ejemplo:

```
int A = M[0][1] + M[1][1];      // A = 7 + 6 = 13
int B = 2 + M[1][2];            // B = 2 + 4 = 6
M[0][0] = A + B;                // M[0][0] = 13 + 6 = 19
```

2.6 Declaración de Matrices en JAVA

Los arreglos ocupan espacio en la memoria. El programador especifica el **tipo** de los elementos y usa el operador **new** para asignar espacio de almacenamiento al número de elementos requerido para arreglo. Entonces para declarar la matriz **M** de los ejemplos anteriores sería:



En JAVA una vez creado un vector con datos numéricos los datos del vector por defecto se inicializan en **cero**

M =

0	0	0	0
0	0	0	0
0	0	0	0

2.7 Ejercicios con matrices

Programa para insertar datos por teclado en una matriz de 3 x 3 y posteriormente visualiza los datos de la matriz

```
import java.util.*;
class matriz
{
   public static void main(String args[])
   {
      Scanner EN=new Scanner(System.in);
      int M[][]=new int[3][3];

      for(int i=0;i<3;i++)
         for(int j=0;j<3;j++)
            M[i][j]=EN.nextInt();

      for(int i=0;i<3;i++)
      {
         for(int j=0;j<3;j++)
            System.out.print(M[i][j]+" ");
         System.out.println();
      }
   }
```

Declaración de la matriz M de 3 x 3

2 bucles for para insertar por teclado los datos a la matriz

2 bucles for para visualizar los datos de la matriz

```
    }
```

Programa para insertar datos por teclado en una matriz de M x N (M y N introducidos por teclado) y posteriormente visualizar los datos de la matriz

```
import java.io.*;
class matriz
{
   public static void main(String args[])
   {
      Scanner EN=new Scanner(System.in);
      int mat[][]=new int[10][10];
      int M,N;

      System.out.println("inserte la cantidad de filas");
      M=EN.nextInt();

      System.out.println("inserte la cantidad de columnas");
      N=EN.nextInt();

      System.out.println("inserte los datos");

      for(int i=0;i<M;i++)
         for(int j=0;j<N;j++)
            mat[i][j]=EN.nextInt();

      for(int i=0;i<M;i++)
      {
         for(int j=0;j<N;j++)
            System.out.print(mat[i][j]+" ");
         System.out.println();
      }
   }
}
```

M y N por teclado