

Estructuras de Datos #3

Arreglos Parte 1

Objetivos

- Describir la estructura de datos tipo arreglo. Uso de vectores y matrices.
- Desarrollo de soluciones con uso de vectores

ARREGLOS

- Los arreglos son estructuras de datos que consisten en elementos del mismo tipo. Los arreglos son entidades estáticas en cuanto a que su tamaño no cambia una vez que han sido creadas.
- Se debe tener en cuenta que antes de usar los elementos de un arreglo, se debe declarar, crear e inicializar el arreglo.

ARREGLOS

- Arreglos unidimensionales o **vectores**

Representación:

números

0	1	2	3
32	-2	0	174

tamaño=4

- Arreglos bidimensionales o **matrices**

Representación:

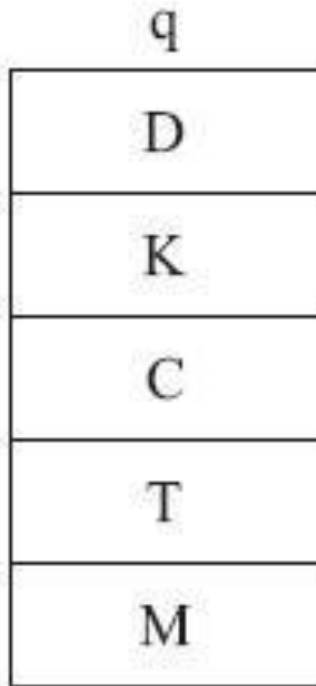
números

	0	1	2	3
0	32	-2	0	174
1	14	12	-5	129

filas=2, columnas=4

Almacenamiento de un Arreglo

- Un arreglo es una colección de datos de un mismo tipo
- Se almacena linealmente en memoria:



Arreglos Tipo Vector

DECLARACIÓN:

En Pseudocódigo

tipoDato nombreVector []

Ejemplos;

- TEXTO nombres []
- NUMERO sueldos []
- CARACTER letras []

En Java

tipoDato nombreVector [] ;

tipoDato [] nombreVector ;

Ejemplos;

- String nombres [] ;
- double sueldos [] ;
- char letras [] ;

Arreglos Tipo Vector

CREACIÓN:

En Pseudocódigo

CREAR nombreVector [tamaño]

Ejemplos;

- CREAR nombres [10]
- CREAR sueldos [numNot]
- CREAR letras [LONGITUD texto]

En Java

nombreVector = new tipoDato [tamaño] ;

Ejemplos;

- nombres = new String [10] ;
- sueldos = new double [numNot] ;
- letras = new char [texto.length()] ;

Arreglos Tipo Vector

INGRESO DE DATOS:

En Pseudocódigo

nombreVector [índice] = valor

Ejemplos;

- nombres [0] = "Danae"
- sueldos [1] = 100.5
- letras [x] = 'J'

En Java

identificador [índice] = valor ;

Ejemplos;

- nombres [0] = "Danae" ;
- sueldos [1] = 100.5 ;
- letras [x] = 'J' ;

Cuando utilizamos vectores podemos utilizar las estructuras lógicas de repetición para optimizar de manera significativa el proceso de ingreso de datos especialmente utilizando la estructura DESDE ó PARA.

Arreglos Tipo Vector

TAMAÑO:

En Pseudocódigo

variable = LONGITUD nombreVector []

Ejemplos;

- tamaño = LONGITUD nombres []
- tamaño = LONGITUD sueldos []
- tamaño = LONGITUD letras []

En Java

variable = nombreVector.length ;

Ejemplos;

- tamaño = nombres.length ;
- tamaño = sueldos.length ;
- tamaño = letras.length ;

Arreglos Tipo Vector

DECLARACIÓN Y CREACIÓN:

En Java

```
tipoDato nombreVector [ ] = new tipoDato [tamaño] ;
```

DECLARACIÓN, CREACIÓN E INICIALIZACIÓN:

En Java

```
tipoDato nombreVector [ ] = { valor1, valor2, valor3, valor4, ... , valorN } ;
```

Arreglos Tipo Vector

- Ejercicio 1
- Desarrollar una solución que permita ingresar y almacenar las notas de todas las prácticas de un alumno. Determinar el promedio de las prácticas sabiendo que se elimina la práctica con la nota mas baja.
- Utilice vectores.

Algoritmo
VectorNotas

ENTRADA: numPra, notas[] / SALIDA: promAlu

INICIO

ENTERO i, numPra

REAL sumaNotas = 0, notaMenor = 20

REAL notas[], promAlu

COMENTARIO "Creación del arreglo"

CREAR notas [numPra]

COMENTARIO "Ingreso de datos al arreglo notas[]."

PARA (i=0, i<numpra, i=i+1) HACER

LEER notas[i]

FINPARA

COMENTARIO "Cálculo del promedio de notas de las prácticas"

PARA (i=0, i<numpra, i=i+1) HACER

sumaNotas = sumaNotas + notas[i]

FINPARA

PARA (i=0, i<numpra, i=i+1) HACER

SI (notaMenor > notas[i]) ENTONCES

notaMenor = notas[i]

FINSI

FINPARA

promAlu = (sumaNotas - notaMenor) / (numPra - 1)

ESCRIBIR promAlu

```

public class PrgVectorNotas {
    public static void main(String[] args) {
        // Declaración de variables.
        int i, numPra ;
        double sumaNotas = 0, notaMenor = 20, promAlu ;
        // Determinación del tamaño del arreglo
        System.out.print ( " Ingrese el número de prácticas del alumno : " ) ;
        numPra = Lectura.leerInt( ) ;
        // Declaración y creación del arreglo.
        double notas[ ] = new double [numPra] ;
        // Ingreso de datos al arreglo notas[ ]
        for ( i = 0 ; i < numPra ; i++ ) {
            System.out.print ( " Ingrese la nota de la práctica [ " + ( i + 1 ) + "]: " ) ;
            notas[i] = Lectura.leerDouble( ) ;
        }
        // Cálculo del promedio de notas de las prácticas.
        for ( i = 0 ; i < numPra ; i++ ) {
            sumaNotas = sumaNotas + notas[i] ;
        }
        for ( i = 0 ; i < numPra ; i++ ) {
            if ( notaMenor > notas [i] ){
                notaMenor = notas[i] ;
            }
        }
        promAlu = (sumaNotas-notaMenor) / (numPra-1) ;
        System.out.println ( " El promedio es : " + promAlu ) ;
    }
}

```

Arreglos tipo MATRIZ

Declaración:

En Pseudocódigo

tipoDato *identificador* [][]

Ejemplos;

- **TEXTO** *nombres* [][]
- **NUMERO** *sueldos* [][]
- **CARACTER** *letras* [][]

En Java

tipoDato *identificador* [][] ;
ó
tipoDato [][] *identificador* ;

Ejemplos;

- **String** *nombres* [][] ;
- **double** *sueldos* [][] ;
- **char** *letras* [][] ;

Arreglos Tipo Matriz

Creación:

En Pseudocódigo

Ejemplos;

- **CREAR** *nombres* [2][4]
- **CREAR** *sueldos* [numAlu][numNot]
- **CREAR** *letras* [4][total]

En Java

identificador = new **tipoDato** [filas][columnas] ;

Ejemplos;

- *nombres* = new **String** [2][10] ;
- *sueldos* = new **double** [numAlu][numNot] ;
- *letras* = new **char** [4][total] ;

Arreglos Tipo Matriz

Declaración y Creación:

En Java

```
tipoDato identificador[][] = new tipoDato [filas][columnas] ;
```

Ejemplos;

- `String nombres = new String [2][10] ;`
- `double sueldos = new double [numAlu][numNot] ;`
- `char letras = new char [4][total] ;`

Arreglos Tipo Matriz

Ingreso de datos:

En Pseudocódigo

identificador [índice] = valor

Ejemplos;

- *nombres* [0][0] = "Danae"
- *sueldos* [1][x] = 100.5
- *letras* [y][x] = 'J'

En Java

identificador [índice] = valor ;

Ejemplos;

- *nombres* [0][0] = "Danae" ;
- *sueldos* [1][x] = 100.5 ;
- *letras* [y][x] = 'J' ;

Cuando utilizamos matrices podemos utilizar las estructuras lógicas de repetición para optimizar de manera significativa el proceso de ingreso de datos especialmente utilizando la estructura *DESDE* ó *PARA* dos veces.

Arreglos Tipo Matriz

Ejercicio

Desarrollar una solución que permita ingresar y almacenar números enteros positivos en una matriz de 3 por 3 (`a[][]`) y permita mostrar la matriz 3 por 3 y la suma de los números ubicados en la diagonal principal. Consistencia la entrada de datos.

Utilice matrices.

ENTRADA: numero[][] / SALIDA: numero[][], sumaDiaPri

INICIO

COMENTARIO "Declaración del arreglo"

ENTERO i, j, numero[][], sumaDiaPri

COMENTARIO "Creación del arreglo"

CREAR numero[3][3],

COMENTARIO "Ingreso de datos al arreglo numero[][]."

PARA (i=0, i<3, i=i+1) HACER // DESDE i =0 HASTA i < 3

PARA (j=0, j<3, j=j+1) HACER // DESDE j =0 HASTA j < 3

HACER

LEER numero[i][j]

MIENTRAS (numero[i][j]<0)

FINPARA

// FIN DESDE

FINPARA

// FIN DESDE

COMENTARIO "Mostrar Matriz y Cálculo de Suma"

PARA (i=0, i<3, i=i+1) HACER // DESDE i =0 HASTA i < 3

PARA (j=0, j<3, j=j+1) HACER // DESDE j =0 HASTA j < 3

ESCRIBIR numero[i][j]

SI (i=j) ENTONCES

sumaDiaPri = sumaDiaPri + numero[i][j]

FINSI

FINPARA

// FIN DESDE

FINPARA

// FIN DESDE

ESCRIBIR sumaDiaPri

FIN

```

public class PrgMatrizDiagonal {
    public static void main(String[] args) {
        // Declaración de variables - Declaración y creación del arreglo bidimensional.
        int i,j, numero[][] = new int [3][3], sumaDiaPri = 0;
        // Ingreso de datos al arreglo notas[ ]
        for(i=0;i<3;i++){
            for(j=0;j<3;j++){
                do{
                    System.out.println("Ingresar Número Posición "+(i+1)+(j+1));
                    numero[i][j]=Lectura.leerInt();
                }while(numero[i][j]<0);
            }
        }
        // Mostrar Matriz y Calcular Suma de Diagonal Principal.
        for(i=0;i<3;i++){
            for(j=0;j<3;j++){
                System.out.print(numero[i][j]+"\\t");
                if(i==j){
                    sumaDiaPri=sumaDiaPri+numero[i][j];
                }
            }
            System.out.println("");
        }
        // Mostrar Suma de Diagonal Principal.
        System.out.println("la Suma de la Diagonal principal es: "+sumaDiaPri);
    }
}

```