Programación C#

Estructura de Datos Tipo Vector

Hemos empleado variables de distinto tipo para el almacenamiento de datos (variables int, string).

En esta unidad veremos otros tipos de variables que permiten almacenar un conjunto de datos en una única variable.

Un vector es una estructura de datos que permite almacenar un CONJUNTO de datos del MISMO tipo.

Con un único nombre se define un vector y por medio de un subíndice hacemos referencia a cada elemento del mismo (componente).

Ejercicio 1

Se desea guardar los sueldos de 5 operarios. Según lo conocido deberíamos definir 5 variables si queremos tener en un cierto momento los 5 sueldos almacenados en memoria.

Empleando un vector solo se requiere definir un único nombre y accedemos a cada elemento por medio del subíndice.

sueldos

1200	750	820	550	490
sueldos[0]	sueldos[1]	sueldos[2]	sueldos[3]	sueldos[4]

Vector (Tamaño de un vector)

Como hemos visto cuando se crea un vector indicamos entre corchetes su tamaño:

```
sueldos=new int[5];
```

Luego cuando tenemos que recorrer dicho vector disponemos una estructura repetitiva for:

```
for(int f=0;f<5;f++)
{
    Console.Write("Ingrese valor de la componente:");
    string linea;
    linea=Console.ReadLine();
    sueldos[f]=int.Parse(linea);
}
```

Como vemos el for se repite mientras el contador f vale menos de 5. Este estructura repetitiva es idéntica cada vez que recorremos el vector.

Que pasa ahora si cambiamos el tamaño del vector cuando lo creamos:

```
sueldos=new int[7];
```

Vector (Tamaño de un vector)

Con esto tenemos que cambiar todos los for que recorren dicho vector. Ahora veremos que un vector al ser un objeto tiene una propiedad llamada Length que almacena su tamaño. Luego podemos modificar todos los for con la siguiente sintaxis:

```
for(int f=0;f<sueldos.Length;f++)
{
    Console.Write("Ingrese valor de la componente:");
    string linea;
    linea=Console.ReadLine();
    sueldos[f]=int.Parse(linea);
}
```

También podemos pedir al usuario que indique el tamaño del vector en tiempo de ejecución, en estos casos se hace imprescindible el empleo de la propiedad Length.

Ejercicio 2

Realizar un programa donde almacene los sueldos de operarios. Cuando se ejecute el programa se debe pedir la cantidad de sueldos a ingresar. Luego crear un vector con dicho tamaño.

Vectores Paralelos

Este concepto se da cuando hay una relación entre las componentes de igual subíndice (misma posición) de un vector y otro.

Si tenemos dos vectores de 5 elementos cada uno. En uno se almacenan los nombres de personas en el otro las edades de dichas personas.

Decimos que el vector nombres es paralelo al vector edades si en la componente 0 de cada vector se almacena información relacionada a una persona (Juan - 12 años).

Es decir hay una relación entre cada componente de los dos vectores. Esta relación la conoce únicamente el programador y se hace para facilitar el desarrollo de algoritmos que procesen los datos almacenados en las estructuras de datos.

Ejercicio 3

Desarrollar un programa que permita cargar 5 nombres de personas y sus edades respectivas. Luego de realizar la carga por teclado de todos los datos notificar los nombres de las personas mayores de edad (mayores o iguales a 18 años)

Vectores (mayor y menor elemento) Ejercicio 4

Desarrollar un programa que permita cargar los nombres de 5 operarios y sus sueldos respectivos. Mostrar el sueldo mayor y el nombre del operario.)

Vectores (ordenamiento)

El ordenamiento de un vector se logra intercambiando las componentes de manera que:

vec[0] <= vec[1] <= vec[2] etc.

El contenido de la componente vec[0] sea menor o igual al contenido de la componente vec[1] y así sucesivamente.

Si se cumple lo dicho anteriormente decimos que el vector está ordenado de menor a mayor. Igualmente podemos ordenar un vector de mayor a menor.

Se puede ordenar tanto vectores con componentes de tipo int, float como string. En este último caso el ordenamiento es alfabético

Ejercicio 5

Desarrollar un programa y definir un vector donde almacenar los nombres de 5 paises. Confeccionar el algoritmo de ordenamiento alfabético.

Vectores (ordenamiento con vectores paralelos)

Cuando se tienen vectores paralelos y se ordena uno de ellos hay que tener la precaución de intercambiar los elementos de los vectores paralelos.

Ejercicio 6

Desarrollar un programa que permita cargar los nombres de 5 alumnos y sus notas respectivas. Luego ordenar las notas de mayor a menor. Imprimir las notas y los nombres de los alumnos.

Matriz

EUna matriz es una estructura de datos que permite almacenar un CONJUNTO de datos del MISMO tipo.

Con un único nombre se define la matriz y por medio de DOS subíndices hacemos referencia a cada elemento de la misma (componente).

EJEMPLO EN EXCEL

En este ejemplo almacenamos valores enteros. Todos los elementos de la matriz deben ser del mismo tipo (int, float, string etc.)
Las filas y columnas comienzan a numerarse a partir de cero, similar a los vectores.

Ejercicio 7

Crear una matriz de 3 filas por 5 columnas con elementos de tipo int, cargar sus componentes y luego imprimirlas.

Ejercicio 8

Crear y cargar una matriz de 3 filas por 4 columnas. Imprimir la primer fila. Imprimir la última fila e imprimir la primer columna.

Matrices (cantidad de filas y columnas)

Como hemos visto para definir y crear la matriz utilizamos la siguiente sintaxis: int[,] mat;

Creación: mat=new int[3,4];

Como las matrices son objetos en C# disponemos de un método llamado GetLength que le pasamos como parámetro la dimension y nos retorna el valor de dicha dimensión.

Si queremos saber la cantidad de filas que tiene la matriz debemos llamar al método GetLength con el valor cero:

```
Console.WriteLine("Cantidad de filas de la matriz:" + mat.GetLength(0));
```

Si queremos saber la cantidad de columnas luego:

```
Console.WriteLine("Cantidad de columnas de la matriz:" + mat.GetLength(1));
```

La primer dimensión son la cantidad de filas y la segunda dimensión son la cantidad de columnas de la matriz.

Ejercicio 9

Crear una matriz de n * m filas (cargar n y m por teclado) Imprimir la matriz completa y la última fila.

Matrices y vectores paralelos

Dependiendo de la complejidad del problema podemos necesitar el empleo de vectores y matrices paralelos.

Ejercicio 10

Se tiene la siguiente información:

· Nombres de 4 empleados.

· Ingresos en concepto de sueldo, cobrado por cada empleado, en los últimos 3 meses.

Confeccionar el programa para:

a) Realizar la carga de la información mencionada.

b) Generar un vector que contenga el ingreso acumulado en sueldos en los últimos 3 meses para cada empleado.

c) Mostrar por pantalla el total pagado en sueldos a todos los empleados en los últimos 3 meses

d) Obtener el nombre del empleado que tuvo el mayor ingreso acumulado

Constructor de la clase

En C# podemos definir un método que se ejecute inicialmente y en forma automática. Este método se lo llama constructor.

El constructor tiene las siguientes características:

- Tiene el mismo nombre de la clase.
- Es el primer método que se ejecuta.
- Se ejecuta en forma automática.
- No puede retornar datos.
- Se ejecuta una única vez.
- Un constructor tiene por objetivo inicializar atributos.

Ejercicio 11

Se desea guardar los sueldos de 5 operarios en un vector. Realizar la creación y carga del vector en el constructor.