

Manejo de Memoria Estática y Dinámica.

¿Qué es Memoria? Es un espacio lógico para guardar información.

¿Qué es Memoria estática? Es cuando no puede modificarse en tiempo de ejecución, es decir, la asignación de memoria puede hacerse en tiempo de compilación y los objetos están vigentes desde que comienza la ejecución del programa hasta que termina. Es la utilizada por variables globales y las declaradas de tipo static. Estos objetos tienen asignada la misma dirección de memoria desde el comienzo hasta el final del programa. (variables globales, tipo static)

La memoria estática es la que se reserva al momento de compilación antes de comenzar a ejecutarse el programa. Los objetos son creados en ese momento y destruidos al final del programa. Mantiene la misma de localización en memoria durante todo el transcurso del programa.

En resumen el inconveniente de utilizar memoria estática aunque es más fácil de programar es que la cantidad de memoria se reserva siempre antes de conocer los datos completos del problema lo que a veces lleva a reservar un máximo de memoria que en la mayoría de las veces no se va a necesitar.

Los objetos administrados de este modo son:

- Variables Static.
- Variables Globales.
- Miembros Static de la Clase.
- Literales de cualquier tipo.

Consideraciones

- Error en tiempo de ejecución de índice fuera de rango
- Se debe conocer con anticipación el tamaño de la estructura
- Se guardan en memoria adyacente
- Vectores, Matrices, Cubos, Registros, Archivos

Ventajas

- Son más fáciles de diseñar.
- Lógica simple
- Óptimas para resolver problemas

Desventajas

- No se puede modificar el tamaño de la estructura en T.D.E
- No es óptima con grandes cantidades de datos
- Desperdicios de memoria cuando no se utiliza totalidad del tamaño

Class CSimple1

{

static void Main(string[] args)

{

```

int[] Numeros=new int[]{1,2,3,4,5};
for(int i=0;i<5;i++)
console.write("{0},",Numeros[i]);
}}
Class CSimple2
{
static int Funcion(int p,int q)
{
return(p+q);
}
static void main(string[] args)
{
int Resultado=Funcion(7,2);
console.writeline(Resultado);
}
}

```

¿Qué es Memoria Dinámica? Es también llamado almacenamiento libre porque en este caso el programador es el que solicita memoria para crear los objetos y es el responsable de liberar la memoria cuando ya no la necesita para ser reutilizada.

Que se modifica permanentemente, es decir, a medida que el proceso va necesitando espacio para más líneas, va solicitando más memoria al sistema operativo para guardarlas. Si puede modificarse en tiempo de ejecución, se puede aumentar o reducir el tamaño de la estructura sin ningún problema.

La reserva de memoria dinámica se hace en tiempo de ejecución después de leer los datos y de conocer el tamaño exacto del problema a resolver, como consecuencia se adapta mejor a las necesidades de cada caso pero en contrapartida es un poco más difícil de programar.

Tanto la creación como destrucción de los objetos está en manos del programador atreves de los operadores new y delete, el sitio donde se almacenan los objetos suele denominarse heap o free store traducido como montículo o memoria libre.

Consideraciones

- Manejo transparente del tamaño de la estructura
- Se guarda donde encuentra espacio
- No se guardan necesariamente en memoria adyacente

Ventajas

- Mayor densidad y capacidad.
- Menor costo por bit.
- Menor consumo de potencia.

Desventajas

- Necesita recargar de la información. almacenada para retenerla.
- Diseño complejo.

```
class Program
{
    void imprime_binario(int n)
    {
        If(n>=2)
        {
            imprime_binario(n/2);
            Console.write("{0}",n%2);
        }
        else
        {
            console.write("{0}",n);
        }
    }

    static void Main(string[] args)
    {
        console.write("Alimenta un numero entero");
        Int numero=int32.Parse(console.readline());
        Program p=new Program();
        console.write("Numero Entero {0} en codigo binario:",numero);
        p.imprime_binario(numero);
        console.readline();
    }
}
```

Bibliografías:

<http://diego-estructuradedatos1.blogspot.com/2011/04/memoria-estatica-y-dinamica.html>

<http://saibercaero.blogspot.com/2011/03/memoria-estatica-y-dinamica.html>

<http://iscestructuradedatosjesusolivas.blogspot.com/2008/09/manejo-de-memoria-estatica.html>

<http://apuntesestructuradedatos.blogspot.com/2009/03/unidad-ii-manejo-de-memoria.html>