

PROGRAMACIÓN

Guía de estudio 11: Te doy una lista de los arreglos que faltan hacer

Listas de arreglos / arreglos dinámicos en C#

Nivel: Secundario - Modalidad Educación Técnico-Profesional.

Ciclo: Segundo ciclo.

Especialidades: Informática, Programación.

Introducción

Como bien saben, para utilizar un arreglo es necesario declararlo, y en esa declaración tenemos que indicar cuántas posiciones debe tener. Pero ¿si no sabemos cuántas posiciones necesitamos? Para eso existen los Array List. Estos no van a permitir agregar elementos sin tener un "tope" de elementos máximos.

¿Qué estamos aprendiendo? Cómo crear un arreglo dinámico, agregar elementos, eliminarlos y recorrerlos.

Materiales de Estudio

Video nro 24 del curso:
<https://youtu.be/LDGtho4ncuw>



Secuencia didáctica

Mirá el siguiente video para aprender cómo crear estos Array List. En el video explica cómo crearlos y agregar elementos. Como esto no alcanza, dejamos algunas notas en los ejercicios para aprender cómo eliminarlos o saber cuántos elementos tienen.

<https://youtu.be/LDGtho4ncuw>



Actividades

Después de mirar el video del link o del QR, te invitamos a que reflexiones con las siguientes preguntas.

No olvides escribir tus respuestas, te facilitarán realizar trabajos prácticos y repasar los temas que hayas estudiado con esta guía.

- 1) ¿Cuál es la diferencia principal entre un arrayList y un array común (recuerden que array es arreglo en inglés)? ¿Te parece útil? Da un ejemplo de un programa en el cual lo usarías (no hace falta escribir el código, simplemente describí el comportamiento).
- 2) Ya hemos hecho un programa de un "gestor de compras", pero ahora vamos a hacer un "gestor de compras **profesional**". El programa iniciará dando la bienvenida (con un mensajito lindo) y preguntará un producto que se quiera agregar. Agregará productos hasta que se ingrese "0" como nombre de producto. Por último, se imprimirá la lista entera de productos.
NOTA: podés saber cuántos datos tiene un ArrayList usando `nombreArray.Count`;
- 3) Hacé un programa para ayudar a el restaurant "Don Manolo" a organizar las mesas que tienen en el local (¡que no sabemos cuántas son!). Los números de mesa comienzan en 1.
 - a) Si se ingresa la palabra "agregar", se deberá preguntar qué número de mesa fue ocupada (y mostrar error si el número es inválido o si la mesa ya está ocupada).
 - b) Si se ingresa la palabra "liberar", se deberá preguntar qué número de mesa se desea liberar. (También mostrando el error si el número es inválido o si la mesa estaba desocupada).

- c) Si se ingresa "mesas", se deberán mostrar las mesas ocupadas.
- d) Si se ingresa "salir" se deberá cerrar el programa.

Aclaración: El programa seguirá preguntando las opciones hasta que se indique salir.

NOTA: se puede eliminar un elemento usando:

```
nombreArray.remove(elemento a remover).
```

Para seguir aprendiendo...

Vayamos con los datos de color. ¿Sabían que el tipo de dato "String" es también un array?

¡Así es! Es un array de "Char". Char se llama así porque viene de la palabra "character" o caracter en español. De la misma forma que podemos recorrer un array, ¡podemos recorrer un String! Veamos un ejemplo:

Si yo tengo declarada una variable `String hola = "hola";` podemos decir que

```
hola[0] = 'h'  
hola[1] = 'o'  
hola[2] = 'l'  
hola[3] = 'a'
```

Noten que las letras están entre comillas simples (con un solo palito). Para cuando usamos caracteres individuales (en vez de Strings), tenemos que usar este tipo de comillas, si no, no nos va a tomar como un carácter solo.

Además utilizando `largo = hola.Length;` podemos saber cuántos caracteres tiene el String

Sabiendo esto. Te proponemos que hagas una función que detecte cuántas palabras hay en un String usando el siguiente código como base:

```
int cantPalabras(String palabra){  
    return palabras;  
}
```

Consideramos palabra si está separado por espacio, coma, punto, punto y coma, y desde ya, la última palabra.