Eslam Mohamed Abdelfadel \_ S9

namespace C\_\_Practices\_2

{

public class Program

{

class MyList

{

int[] Arr;

int index;

public int Length { get; set; } = 0;

public int Size { get; set; } = 5;

public MyList()

{

Arr = new int[Size];

index = -1;

}

bool IsFull()

{

return index == Size - 1; // 4

}

public void Add(int Element)

{

if (IsFull())

ExtendSizeAndAdd(Element);

else

Arr[++index] = Element; // index = 0 1 2 3 4

Length++;

}

public void ExtendSizeAndAdd(int NewElement)

{

int NewSize = Size \* 2;

int[] NewArr = new int[NewSize];

index = -1;

foreach (int item in Arr)

{

NewArr[++index] = item;

}

Arr = NewArr;

Size = NewSize;

Arr[++index] = NewElement;

}

public int this[int I]

{

get

{

return Arr[I];

}

set

{

Add(value);

}

}

}

static void Main(string[] args)

{

MyList list = new MyList();

list.Add(1);

list.Add(2);

list.Add(3);

list.Add(4);

list.Add(5);

Console.WriteLine(list.Size); // 5

list.Add(6);

Console.WriteLine(list.Size); // 10

list.Add(7);

list.Add(8);

list.Add(9);

list.Add(10);

Console.WriteLine(list.Size); // 10

list.Add(11);

Console.WriteLine(list.Size); // 20

for (int I = 0; I < list.Length; I++)

{

Console.WriteLine(list[I]);

}

}

}

}

The difference between nameof and ToString

· nameof: Used to obtain the name of a variable, type, or member as a string **at compile time**.

· ToString: Converts an object to its string representation **at runtime**.

Who is better?

Nameof is better than ToString, because

Type Safty

Nameof is Type safty ( is checked by the compiler )

ToString Not type safty

Performance

Nameof is faster ( there is no runtime overhead )

ToString slower ( executed at runtime )

Why use nameof?

**Refactor-friendly**: If you rename a variable, method, or class, nameof will automatically reflect the new name.

**Type safety**: The compiler ensures the symbol exists, and you'll get a compile-time error if it doesn't.

Why use ToString?

* **For runtime data**: When you need the actual string representation of an object.
* **Custom representations**: Many classes override the ToString method to provide meaningful representations of their instances.