PROGRAMMING
ADVICES
LEARN THE RIGHT WAY
26+ Years of Experience
Mohammed Abu-Hadhoud
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD
لا تنسى الاشتراك في قناتنا على اليوتيوب ومشاركة القناة مع اصدقائك
لتعم الفائدة للجميع وانقاذ الاف الناس من التشتت وجزاكم الله خيرا
لا تنسونا من دعائكم وادعو لوالدي بالرحمة
www.ProgrammingAdvices.com

مهم جداً

هذا الملف للمراجعة السريعة واخذ الملاحظات عليه فقط ،لانه يحتوي
على اقل من 20٪ مما يتم شرحه في الفيديوهات
الاستعجال والاعتماد عليه فقط سوف يجعلك تخسر كميه معلومات
وخبرات كثيره

يجب عليك مشاهدة فيديو الدرس كاملا

لاتنسى عمل لايك ومشاركة القناة لتعم الفائدة للجميع
لا تنسونا من دعائكم

ProgrammingAdvices.com

Mohammed Abu-Hadhoud

Data Structures Level 2

What is IList Interfaces?

Mohammed Abu-Hadhoud

MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD

ProgrammingAdvices.com

PROGRAMMING ADVICES LEARN THE RIGHT WAY

What is IList Interface?

- IList is an interface that resides in the System.Collections namespace and extends ICollection.
- It represents a collection of objects that can be individually accessed by index, offering a more flexible way to interact with collections.
- The primary advantage of IList is its support for indexed access, which allows for the retrieval, update, or removal of elements at specific positions within the collection.This feature is crucial for many data manipulation scenarios where order and position matter.

Key Features of IList

- Index-based access: IList provides the ability to access, modify, or remove items based on their index in the collection.
- Insert and RemoveAt: Add or remove elements at a specified index, adjusting the collection accordingly.
- IndexOf: Find the index of a specific element in the collection.
- Count and IsReadOnly properties: Similar to ICollection, these properties provide information about the size of the collection and whether it is read-only.
- Add, Insert, Remove, and RemoveAt methods: Beyond the capabilities inherited from ICollection, IList allows for inserting and removing items at specified indices.

Best Practices

- Choosing between IList and other collection interfaces: Use IList when you need both sequential access and the ability to manipulate the collection by index.
- If you only need sequential access without modifications, IEnumerable might be sufficient. For collections that require key-value pair management, consider IDictionary.
- Performance considerations: Operations that involve indexing, like inserting or removing at a specific index, can have different performance characteristics depending on the underlying collection type (e.g., List<T> vs. LinkedList<T>). Choose the appropriate concrete collection type based on your performance requirements.

Mohammmed Abu-Hadhoud

26+ Years of Experience

MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD

PROGRAMMING ADVICES LEARN THE RIGHT WAY