Bài: Tổng quan về Collection trong C#

Xem bài học trên website để ủng hộ Kteam: Tổng quan về Collection trong C#

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Đây là bài đầu tiên trong khóa LẬP TRÌNH C# NÂNG CAO. Trong bài này chúng ta sẽ cùng tìm hiểu **Tổng quan về Collections** trong .NET Framework.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- BIẾN , KIỂU DỮ LIỆU, TOÁN TỬ trong C#
- CÂU ĐIỀU KIỆN trong C#
- Cấu trúc cơ bản của VÒNG LẶP, HÀM trong C#
- MẢNG trong C#
- LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG C#

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Collections là gì?
- Một số loại Collection thông dụng.

Collections là gì?

Collections trong C# là gì?

Các lớp hỗ trợ lưu trữ, quản lý và thao tác với các đối tượng một cách có thứ tự.

Các lớp này nằm trong namespace System.Collections.

Một số đặc điểm của Collections

Là một mảng có kích thước động:

- Không cần khai báo kích thước khi khởi tạo.
- Có thể tăng giảm số lượng phần tử trong mảng một cách linh hoạt.

Có thể lưu trữ một tập hợp đối tượng thuộc nhiều kiểu khác nhau.

Hỗ trợ rất nhiều phương thức để thao tác với tập hợp như: tìm kiếm, sắp xếp, đảo ngược, . . .

Mỗi collections được tổ chức thành một lớp nên cần khởi tạo đối tượng trước khi sử dụng.

Khi nào sử dụng Collection?

Chúng ta đã từng tìm hiểu một kiểu dữ liệu dùng để quản lý danh sách đối tượng đó là kiểu mảng. Vậy Collections có gì hay hơn mảng? Khi nào dùng mảng và khi nào dùng Collections?

Đầu tiên là những điểm mạnh của **Collections**

- Bên trong **Collections** có nhiều lớp đa dạng hỗ trợ cho từng mục đích khác nhau. Nếu như mảng chỉ có thể truy xuất phần tử thông qua chỉ số thì các Collections có thể truy xuất thông qua chỉ số hoặc thông qua key.
- Đối với danh sách cần thao tác tìm kiếm nhiều thì Collections cũng có lớp hỗ trợ giúp việc tìm kiếm nhanh hơn nhiều so với mảng nguyên thuỷ (sẽ được trình bày trong bài HASHTABLE TRONG C#).
- Trong trường hợp danh sách cần thay đổi số lượng phần tử liên tục (thêm hoặc xoá phần tử) thì Collections cũng hỗ trợ sẵn (sẽ được trình bày trong bài ARRAYLIST TRONG C#).
- Ngoài ra trong namespace System.Collections còn hỗ trợ sẵn 2 cấu trúc dữ liệu kinh điển đó là STACK và QUEUE nên bạn chỉ cần lấy ra sử dụng mà không cần cài đặt lại.

Vậy khi nào sử dụng mảng khi nào sử dụng Collections?

Theo lời khuyên từ Microsoft thì:

- Mảng thường được dùng để làm việc với một số lượng cố định các đối tượng **strongly-typed** (Mình sẽ trình rõ hơn về khái niệm này trong series khác. Bạn có thể hiểu đơn giản, **strongly-typed** là kiểu dữ liệu không bị thay đổi một cách đột ngột, tường minh).
- Collections cung cấp một cách linh hoạt hơn để làm việc với danh sách. Ta có thể tăng giảm số lượng phần tử một cách tự động. Một số Collections còn hỗ trợ lưu trữ danh sách dưới dạng **Key – Value** (sẽ được trình bày trong bài HASHTABLE TRONG C#) giúp mình truy xuất, tìm kiếm một cách nhanh chóng.

Một số Collections thông dụng

Một số lớp Collections được sử dụng phổ biến:

LỚP	MÔ TẢ
ArrayList	Lớp cho phép lưu trữ và quản lý các phần tử giống mảng. Tuy nhiên, không giống như trong mảng, ta có thể thêm hoặc xoá phần tử một cách linh hoạt và có thể tự điều chỉnh kích cỡ một cách tự động.
HashTable	Lớp lưu trữ dữ liệu dưới dạng cặp **Key – Value**. Khi đó ta sẽ truy xuất các phần tử trong danh sách này thông qua **Key** (thay vì thông qua chỉ số phần tử như mảng bình thường).
SortedList	Là sự kết hợp của ArrayList và HashTable. Tức là dữ liệu sẽ lưu dưới dạng **Key – Value**. Ta có thể truy xuất các phần tử trong danh sách thông qua **Key** hoặc thông qua chỉ số phần tử. Đặc biệt là các phần tử trong danh sách này luôn được sắp xếp theo giá trị của **Key**.
Stack	Lớp cho phép lưu trữ và thao tác dữ liệu theo cấu trúc **LIFO** (*Last In First Out*).
Queue	Lớp cho phép lưu trữ và thao tác dữ liệu theo cấu trúc **FIFO** (*First In First Out*).
BitArray	Lớp cho phép lưu trữ và quản lý một danh sách các **bit**. Giống mảng các phần tử kiểu bool với true biểu thị cho bit **1** và false biểu thị cho bit **0**. Ngoài ra **BitArray** còn hỗ trợ một số phương thức cho việc tính toán trên bit.

Trong bài học này mình chỉ giới thiệu tổng quan về **Collections** và một số Collections thông dụng. Những bài học sau chúng ta sẽ đi vào chi tiết về cách sử dụng từng Collections cụ thể.

Kết luận

Nội dung bài này đã giới thiệu đến các bạn về Collections cũng như một số Collections thông dụng.

Bài sau chúng ta sẽ tìm hiểu về ARRAYLIST TRONG C#.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".