# Bài 5.5: Lớp Array

- ✓ Các thuộc tính
- ✓ Các phương thức
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

# Các thuộc tính

- ✓Lớp Array là lớp cung cấp các phương thức cho phép tạo mới, quản lý, tìm kiếm, sắp xếp mảng.
- ✓Đây là lớp cha của mọi mảng trong ngôn ngữ C#.
- ✓Các thuộc tính:
  - ✓ IsFixedSize: trả về true, giá trị chỉ định liệu mảng có kích thước cố định hay không.
  - ✓ IsReadOnly: trả về giá trị true nếu mảng đang xét là mảng chỉ đọc.
  - ✓ IsSynchronized: trả về giá trị true nếu việc truy cập mảng là đồng bộ.
  - ✓ Length: trả về tổng số phần tử của tất cả các chiều trong mảng.
  - ✓ LongLength: trả về giá trị số phần tử của tất cả các chiều trong mảng ở dạng số nguyên 64 bit.
  - ✓ MaxLength: trả về số phần tử tối đa có thể chứa trong mảng.
  - ✓ Rank: trả về số chiều của mảng. Mảng 1 chiều thì Rank sẽ bằng 1. Mảng n chiều thì Rank là n.

# Các phương thức thường dùng

| Phương thức và mô tả |
| --- |
| **public static int BinarySearch(Array, arr, int index, int length, object? value);**<br>Tìm kiếm nhị phân giá trị value trên mảng 1 chiều đã được sắp xếp.<br>Trong đó:<br>- arr: mảng 1 chiều đã sắp xếp chứa các phần tử để tìm kiếm.<br>- index: vị trí bắt đầu tìm kiếm tính theo chỉ số phần tử mảng.<br>- length: độ dài của đoạn phần tử cần tìm.<br>- value: giá trị cần tìm.<br>Trả về giá trị số nguyên >= 0 nếu tìm thấy value trong mảng arr. Ngược lại trả về giá trị âm. |
| **public static void Clear(Array array);**<br>**public static void Clear(Array array, int index, int length);**<br>Xóa nội dung của cả mảng hoặc của một đoạn length phần tử bắt đầu từ vị trí index.<br>Tham số:<br>- array: mảng cần xóa.<br>- index: vị trí bắt đầu xóa.<br>- length: số lượng phần tử cần xóa. |
| **public object Clone();**<br>Tạo một bản sao của mảng hiện thời. |
| **public static void Copy(Array source, Array dest, int length);(1)**<br>**public static void Copy(Array source, int fromIndex, Array dest, int destIndex, int length);(2)**<br>Sao chép length phần tử từ mảng source sang mảng dest từ vị trí fromIndex(nếu chỉ định) trong mảng source tới vị trí destIndex(nếu chỉ định) trong mảng dest.<br>Tham số:<br>- source: mảng nguồn để copy.<br>- dest: mảng đích.<br>- length: số phần tử cần sao chép.<br>- fromIndex: vị trí bắt đầu chép.<br>- destIndex: vị trí pate vào. |

# Các phương thức thường dùng

**public void CopyTo(Array array, int index);**
Sao chép tất cả các phần tử từ mảng hiện tại sang mảng đích.
Tham số:
- array: mảng đích 1 chiều.
- index: vị trí trong mảng arr tại đó dữ liệu được chép vào.

**public static Array CreateInstance(Type t, int length);(1)**
**public static Array CreateInstance(Type t, params int[] length);(2)**
**public static Array CreateInstance(Type t, int length1, int length2);(3)**
Tạo mảng 1 chiều với kích thước length.(1)
Tạo mảng nhiều chiều với kích thước mỗi chiều cho trong mảng length.(2)
Tạo mảng 2 chiều với số hàng, cột cho trong length1, length2.(3)
Tham số:
- t: kiểu của mảng.
- length: số phần tử mảng.
- length1: số hàng.
- Length2: số cột.

Trả về: đối tượng mảng vừa tạo trong đó các phần tử sẽ được khởi tạo giá trị mặc định của kiểu mảng.

**public static T[] Empty<T>();**
Trả về một mảng rỗng.

**public static bool Exists<T> (T[] arr, Predicate<T> match);**
Kiểm tra xem mảng arr có chứa các phần tử thỏa mãn điều kiện của match không.
Tham số:
- T: kiểu của mảng.
- Array: mảng 1 chiều cần xét.
- Match: định nghĩa của biểu thức điều kiện cần kiểm tra.

Trả về true nếu mảng arr chứa 1 hoặc nhiều phần tử thỏa mãn điều kiện trong match.

# Các phương thức thường dùng

| |
|---|
| **public static void Fill<T>(T[] array, T value, int startIndex, int count);(1)**<br>**public static void Fill<T>(T[] arr, T value);(2)**<br>Gán giá trị value cho tất cả các phần tử của mảng arr.<br>(1)Nếu chỉ định startIndex và count, count phần tử từ vị trí startIndex sẽ được gán giá trị value.<br>Tham số:<br>- T: kiểu của các phần tử trong mảng.<br>- arr: mảng cần gán giá trị.<br>- value: giá trị cần gán.<br>- startIndex: vị trí bắt đầu gán.<br>- count: số phần tử cần gán. |
| **public static T? Find<T>(T[] arr, Predicate<T> match);**<br>Tìm phần tử đầu tiên thỏa mãn điều kiện trong match và trả về phần tử đó.<br>Tham số:<br>- T: kiểu của mảng.<br>- arr: mảng cần tìm.<br>- match: điều kiện<br>Trả về: phần tử đầu tiên thỏa mãn điều kiện hoặc giá trị mặc định của kiểu T. |
| **public static T[] FindAll<T>(T[] arr, Predicate<T> match);**<br>Trả về mảng các phần tử thỏa mãn điều kiện chỉ ra bởi match.<br>Tham số:<br>- T: kiểu của mảng.<br>- arr: mảng cần tìm.<br>- match: điều kiện<br>Trả về: mảng chứa các phần tử thỏa mãn điều kiện. Hoặc mảng rỗng nếu không có phần tử nào thỏa mãn. |

# Các phương thức thường dùng

| |
|---|
| **public static int FindIndex<T>(T[] arr, Predicate<T> match); (1)**<br>**public static int FindIndex<T>(T[] arr, int startIndex, Predicate<T> match); (2)**<br>**public static int FindIndex<T>(T[] arr, int startIndex, int count, Predicate<T> match); (3)**<br>(1) Tìm và trả về chỉ số phần tử đầu tiên trong mảng arr thỏa mãn điều kiện trong match.<br>(2) Tìm và trả về chỉ số phần tử đầu tiên trong mảng arr tính từ vị trí startIndex tới cuối mảng thỏa mãn điều kiện trong match.<br>(3) Tìm và trả về chỉ số phần tử đầu tiên trong count phần tử của mảng arr tính từ vị trí startIndex thỏa mãn điều kiện trong match.<br>Tham số:<br>- T: kiểu của mảng.<br>- arr: mảng cần tìm.<br>- match: điều kiện cần thỏa mãn.<br>- startIndex: vị trí bắt đầu tìm.<br>- count: số phần tử cần xét.<br>Trả về vị trí phần tử đầu tiên tìm được hoặc -1 nếu không tìm thấy. |
| **public static T? FindLast<T>(T[] arr, Predicate<T> match);**<br>Tương tự phương thức Find nhưng trả về phần tử cuối cùng tìm được. |
| **public static int FindLastIndex<T>(T[] arr, Predicate<T> match); (1)**<br>**public static int FindLastIndex<T>(T[] arr, int startIndex, Predicate<T> match); (2)**<br>**public static int FindLastIndex<T>(T[] arr, int startIndex, int count, Predicate<T> match); (3)**<br>Tương tự như phương thức FindIndex() nhưng trả về vị trí phần tử cuối tìm được. |
| **public static void ForEach<T>(T[] arr, Action<T> action);**<br>Thực hiện một hành động cụ thể trên từng phần tử của mảng arr.<br>Tham số:<br>- T: kiểu của mảng.<br>- arr: mảng cần xét.<br>- action: hành động cần thực hiện. |

# Các phương thức thường dùng

| |
|---|
| **public int GetLength(int dimension);**<br>Trả về số phần tử của chiều được chỉ định trong mảng arr.<br>Tham số:<br>- dimension: chiều của mảng cần xét. Tính từ 0.<br>Trả về: số phần tử ở chiều được chỉ định. |
| **public int GetLowerBound(int dimension);**<br>**public int GetUpperBound(int dimention);**<br>Lấy chỉ số phần tử đầu tiên và cuối cùng của mảng hiện thời tại chiều được chỉ định.<br>Tham số:<br>- dimension: chiều cần xét. Tính từ 0.<br>Trả về: chỉ số phần tử đầu tiên/cuối cùng tại chiều dimension. |
| **public object? GetValue(int index); (1)**<br>**public object? GetValue(params int[] indice); (2)**<br>**public object? GetValue(int rowIndex, int colIndex); (3)**<br>**public object? GetValue(int index1, int index2, int index3); (4)**<br>(1) Lấy giá trị phần tử tại vị trí index trong mảng 1 chiều.<br>(2) Lấy giá trị phần tử tại vị trí cho trong tham số indice trong mảng đa chiều.<br>(3) Lấy giá trị phần tử tại vị trí hàng, cột cho trước trong mảng 2 chiều.<br>(4) Lấy giá trị phần tử tại vị trí cho trước trong mảng 3 chiều.<br>Tham số:<br>- index, rowIndex, colIndex, indice, index1, index2, index3: vị trí chỉ số tại chiều tương ứng. |

# Các phương thức thường dùng

| |
|---|
| **public static int IndexOf(Array arr, object? value); (1)**<br>**public static int IndexOf(Array arr, object? value, int startIndex); (1)**<br>**public static int IndexOf(Array arr, object? value, int startIndex, int count); (1)**<br>**public static int LastIndexOf(Array arr, object? value); (2)**<br>**public static int LastIndexOf(Array arr, object? value, int startIndex); (2)**<br>**public static int LastIndexOf(Array arr, object? value, int startIndex, int count); (2)**<br>Tìm vị trí chỉ số phần tử đầu tiên(1) hoặc cuối cùng(2) trong mảng có giá trị bằng value.<br>Tham số:<br>- arr: mảng gốc để tìm.<br>- value: giá trị cần tìm.<br>- startIndex: vị trí bắt đầu tìm.<br>Trả về: vị trí phần tử thỏa mãn. |
| **public void Initialize();**<br>Khởi tạo tất cả các phần tử trong mảng kiểu value-type bằng cách gọi phương thức khởi tạo không tham số của kiểu đó. |
| **public static void Resize<T>(ref T[]? Arr, int newSize);**<br>Thay đổi kích thước mảng về newSize phần tử.<br>Nếu arr là null thì phương thức sẽ tạo mới mảng với kiểu T có newSize phần tử. |
| public static void Reverse(Array arr, int index, int length);<br>public static void Reverse(Array arr);<br>public static void Reverse<T>(T[] arr);<br>public static void Reverse<T>(T[] arr, int index, int length);<br>Đảo ngược vị trí các phần tử trong toàn mảng arr hoặc trong đoạn length phần tử từ vị trí index.<br>Tham số:<br>- arr: mảng cần đảo ngược các phần tử.<br>- index: vị trí bắt đầu.<br>- length: số phần tử cần đảo ngược.<br>- T: kiểu của mảng generic. |

# Các phương thức thường dùng

**public void SetValue(object? value, int index); (1)**
**public void SetValue(object? value, params int[] indice); (2)**
**public void SetValue(object? value, int index1, int index2); (3)**
**public void SetValue(object? value, int index1, int index2, int index3); (4)**
Phương thức gán giá trị cho phần tử tại vị trí được chỉ định trong mảng 1 chiều, đa chiều, 2, 3 chiều.

**public static void Sort(Array arr);**
**public static void Sort(Array arr, int index, int length);**
**public static void Sort<T>(T[] arr);**
**public static void Sort<T>(T[] arr, Comparison<T> comparison);**
**public static void Sort<T>(T[] arr, int index, int length);**
Sắp xếp mảng theo tứ tự tăng dần các phần tử hoặc theo điều kiện chỉ ra trong comparison.
Nếu chỉ rõ index và length, phương thức sắp xếp length phần tử trong mảng arr từ vị trí index.
Tham số:

- Arr: mảng cần sắp xếp.
- Index: vị trí phần tử bắt đầu sắp xếp.
- Length: số phần tử cần sắp xếp.
- Comparison: điều kiện sắp xếp.
- T: kiểu của mảng generic.

# Nội dung tiếp theo

## Tìm hiểu về các kiểu giá trị có thể null