

C# - תכנות מכוון עצמים

ממשק interface המשך - מיון

השיטה Array.sort

2

- השיטה Array.sort מקבלת מערך וממיינת את איבריו
- עבור טיפוסים בסיסיים השיטה יודעת לבצע את העבודה מאחר והיחס $<$, $=$, $>$ מוגדרים היטב עבור טיפוסים אלו

```
static void Main(string[] args)
{
    int[] arr = { 4,6,2,8,1 };

    Array.Sort(arr);

    for (int i=0 ; i < arr.Length ; i++)
        Console.Write("{0} ", arr[i]);
    Console.WriteLine();
}
```

```
1 2 4 6 8
Press any key to continue . . . _
```

השיטה Array.sort

עבור מערך אובייקטים

3

```
static void Main(string[] args)
{
    Person[] arr = new Person[3];
    arr[0] = new Person(222, "momo");
    arr[1] = new Person(333, "gogo");
    arr[2] = new Person(111, "koko");
```

```
    Array.Sort(arr);
```

```
    for (int i = 0; i < arr.Length; i++)
        Console.Write("{0} ", arr[i].ToString());
    Console.WriteLine();
}
```

```
class Person
{
    private int id;
    private string name;

    public Person(int id, string name)
    {
        this.id = id;
        this.name = name;
    }

    public override string ToString()
    {
        return "Id: " + id + ", Name: " + name;
    }
}
```

הקומפיילר לא יודע למיין Person...

יש תעופה בניסיון המיין כי השיטה sort
מתבססת על מימוש הממשק Comparable

```
Unhandled Exception: System.InvalidOperationException: Failed to compare two elements in the array. ---> System.ArgumentException: At least one object must implement IComparable.
```

```
at System.Collections.Comparer.Compare(Object a, Object b)
at System.Collections.Generic.ObjectComparer`1.Compare(T x, T y)
at System.Collections.Generic.ArraySortHelper`1.SwapIfGreaterWithItems(T[] keys, IComparer`1 comparer, Int32 a, Int32 b)
at System.Collections.Generic.ArraySortHelper`1.QuickSort(T[] keys, Int32 left, Int32 right, IComparer`1 comparer)
```

השיטה sort מתבססת על הממשק Comparable

4

□ מאחר ו- sort רוצה שהאיברים למיון ידעו לתת תוצאת יחס בין שני איברים (גדול/קטן/שווה), היא מתבססת על כך שהם יממשו את ההתנהגות המוגדרת בממשק הקיים Comparable המכיל את השיטה compareTo

הממשק Comparable

5

□ זהו ממשק המסופק עם השפה, וכדי לממש אותו צריך לממש את השיטה:

public int CompareTo(object obj)

■ השיטה תחזיר מיון בסדר עולה:

■ 0 אם $this \leq obj$ זהים

■ -1 אם $this < obj$

■ 1 אם $this > obj$

□ ממשק זה ממומש כבר במחלקה String

□ נרצה לממש אותו כאשר נרצה לאפשר השוואה בין אובייקטים

□ נעדיף להשתמש בממשק קיים זה ולא להגדיר אותו מחדש כדי לספק שפה אחידה בין המתכנתים

התיקון: מימוש IComparable

6

יש לציין את שם המחלקה,
אחרת תתבצע השוואה מול
Object, מה שיצריך casting

השוואה לפי שדה ה- id

```
class Person : IComparable<Person>
{
    private int id;
    private string name;

    public Person(int id, string name)
    {
        this.id = id;
        this.name = name;
    }

    public override string ToString()
    {
        return "Id: " + id + ", Name: " + name;
    }

    #region IComparable<Person> Members

    public int CompareTo(Person other)
    {
        if (id < other.id) return -1;
        else if (id > other.id) return 1;
        else return 0;
    }

    #endregion
} // class Person
```

התיקון: מימוש IComparable (2)

7

```
static void Main(string[] args)
{
    Person[] arr = new Person[3];
    arr[0] = new Person(222, "momo");
    arr[1] = new Person(333, "gogo");
    arr[2] = new Person(111, "koko");

    Array.Sort(arr);

    for (int i = 0; i < arr.Length; i++)
        Console.WriteLine("{0} ", arr[i].ToString());
    Console.WriteLine();
}
```

```
Id: 111, Name: koko
Id: 222, Name: momo
Id: 333, Name: gogo
```

השוואה לפי

שדה ה-name

8

```
Id: 333, Name: gogo  
Id: 111, Name: koko  
Id: 222, Name: momo
```

```
class Person : IComparable<Person>  
{  
    private int id;  
    private string name;  
  
    public Person(int id, string name)  
    {  
        this.id = id;  
        this.name = name;  
    }  
  
    public override string ToString()  
    {  
        return "Id: " + id + ", Name: " + name;  
    }  
  
    #region IComparable<Person> Members  
    public int CompareTo(Person other)  
    {  
        return name.CompareTo(other.name);  
    }  
  
    #endregion  
} // class Person
```

תרגיל – הרחבת המחלקה Person

9

- הרחבת המחלקה Person שמכילה ת.ז מסוג מספר שלם ושם מסוג מחרוזת
- בצע מיון ראשי ומיון משני בסדר עולה
- המיון הראשי יהיה לפי שם האדם (בסדר עולה)
- במידה והשמות זהים יתבצע מיון משני לפי תעודת הזהות (בסדר עולה)
- דוגמה:

Id: 333, Name: gogo

Id: 111, Name: koko

Id: 222, Name: momo

Id: 444, Name: momo

- ממש פונקציה חיצונית חדשה בתוך המחלקה Program בשם Sort(...) שמחליפה את Array.Sort(...).