



Национална програма  
"Обучение за ИТ умения и кариера"  
<https://it-kariera.mon.bg>

Министерството на  
образованието и науката  
<https://www.mon.bg>



# Компаратори

Comparators

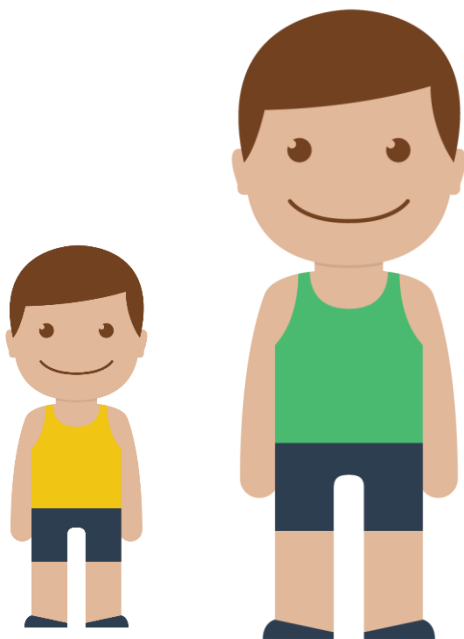
# Съдържание

1. Интерфейс `Comparable<T>`
2. Интерфейс `Comparer<T>`

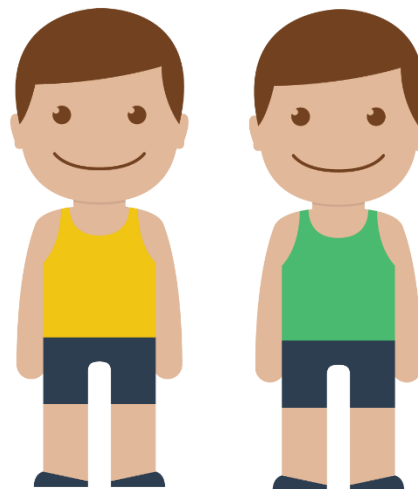
# IComparable<T>

- Да се чете като “Аз съм нещо сравнимо”
- Предоставя метод за сравняване на два обекта от дадения тип – CompareTo()
- Указва подредбата на обектите за сортиране
- Засяга оригиналния клас

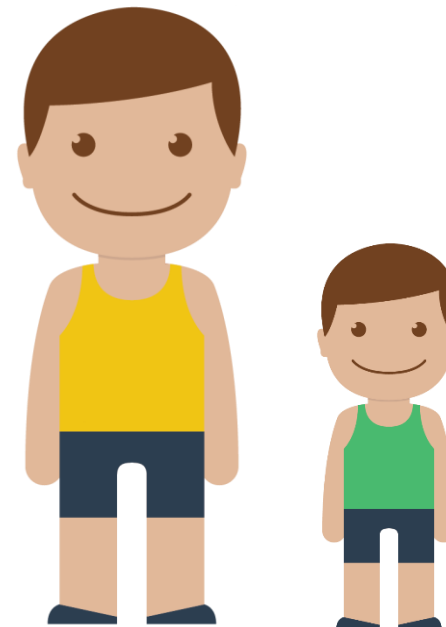
# Резултат, връщан от метода CompareTo(T)



$< 0$



$= 0$



$> 0$

# IComparable<T> npumep

```
class Point : IComparable<Point>
{
    public int X { get; set; }
    public int Y { get; set; }
    public int CompareTo(Point otherPoint)
    {
        if (this.X != otherPoint.X)
            return (this.X - otherPoint.X);

        if (this.Y != otherPoint.Y)
            return (this.Y - otherPoint.Y);

        return 0;
    }
}
```

# IComparable<T> npumep 2

```
class Cat : IComparable<Cat>
{
    public string Name { get; set; }

    public int CompareTo(Cat other)
    {
        return this.Name.CompareTo(other.Name);
    }
}
```

# IComparer<T>

- Да се чете “Аз сравнявам”
- Предоставя начин за **настройване** на **реда** на **сортиране** на **колекция**
- Дефинира **метод**, който даденият тип реализира, за да **сравни два обекта**
- **Не засяга** оригиналния клас

# IComparer<T> npumep

```
class CatComparer : IComparer<Cat>
{
    public int Compare(Cat x, Cat y)
    {
        return x.Name.CompareTo(y.Name);
    }
}
```

```
IComparer<Cat> comparer = new CatComparer();
SortedSet<Cat> catsByName = new SortedSet(comparer);
```



# Задача: Сравнима книга

- Реализирайте интерфейса `Comparable<Book>` за съществуващия клас `Book`.
  - Първо ги сортирайте по **нарастващ хронологичен** ред (по година)
  - Ако две книги са в **една година**, сортирайте ги в **азбучен ред**
- предефинирайте метода `ToString()` за вашия клас `Book` class, така че да връща низ във формат:
  - `{title} - {year}`
- Променете вашия клас `Library`, така че да **съхранява книгите** в **правилната** подредба

# Решение: Сравнума книга

```
public class Book : IComparable<Book>
{
    public int CompareTo(Book other)
    {
        int result = this.Year.CompareTo(other.Year);
        if (result == 0)
        {
            result = this.Title.CompareTo(other.Title);
        }
        return result;
    }
}
```

# Задача: Сравняващият книги

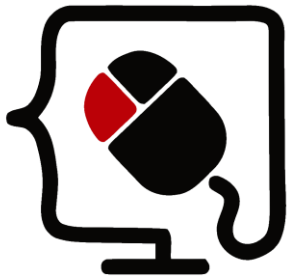
- Създайте клас **BookComparator**, който трябва да реализира интерфейса **IComparer<Book>**
- **BookComparator** трябва да **сравнява** две книги по:
  - Заглавието на книгата – **в азбучен ред**
  - Година на публикуване на книгата – **от най-нови към най-стари**
- Променете вашия клас **Library** още веднъж, така че да реализира новото сортиране

# Решение: Сравнивающий книги

```
public class BookComparator : IComparer<Book>
{
    public int Compare(Book x, Book y)
    {
        int result = x.Title.CompareTo(y.Title);
        if (result == 0)
        {
            result = y.Year.CompareTo(x.Year);
        }
        return result;
    }
}
```

# Обобщение

- Интерфейсът `IComparable<T>` казва „Аз съм нещо сравнимо“ и дава метод за сравняване на два обекта
- Интерфейсът `IComparer<T>` твърди „Аз мога да сравнявам“ и предоставя начин за промяна на реда на сортиране на колекция



Национална програма  
"Обучение за ИТ умения и кариера"  
<https://it-kariera.mon.bg>

Министерството на  
образованието и науката  
<https://www.mon.bg>



Документът е разработен за нуждите на Национална програма "Обучение за ИТ умения и кариера" на Министерството на образованието и науката (МОН) и се разпространява под свободен лиценз CC-BY-NC-SA (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).