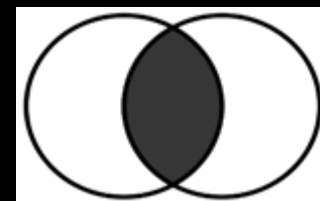


# Алгоритми върху линейни структури от данни. Обединение и сечение

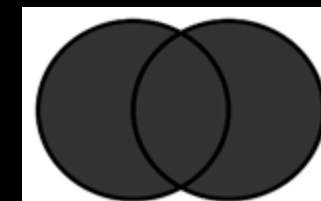
1	2	3	4	5
---	---	---	---	---

2	4	6
---	---	---



сечение

2	4
---	---



обединение

1	2	3	4	5	6
---	---	---	---	---	---



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>

# Съдържание

## 1. Обединение и сечение на списъци

- директно
- с метода **AddRange<T>**



# Обединение и сечение на списъци

- Нека сега разгледаме един по-интересен пример - да напишем програма, която може да намира обединенията и сеченията на две множества числа.
- Можем да приемем, че имаме два списъка и искаме да вземем елементите, които се намират и в двата едновременно (сечение) или търсим тези, които се намират поне в единия от двата (обединение).

1	2	3	4	5
---	---	---	---	---

2	4	6
---	---	---

1	2	3	4	5	6
---	---	---	---	---	---

2	4
---	---

обединение

сечение

# Обединение и сечение на списъци – Вариант 1 пример

- Едно възможно решение е с директно следване на определенията за обединение и сечение на множества
  - Използваме методите "**съдържа ли се**" елемент в списък и "**добави**" елемент към списък
- **Обединение** – това са елементите на първия и тези от втория, които се срещат само в него

```
public static List<int> Union(  
    List<int> firstList,  
    List<int> secondList )  
{  
    List<int> union = new List<int>();  
    union.AddRange(firstList);  
    foreach (var item in secondList)  
    {  
        if (!union.Contains(item))  
        {  
            union.Add(item);  
        }  
    }  
    return union;  
}
```



# Обединение и сечение на списъци – Вариант 1 пример(1)

- **Сечение** – това са елементите на единия списък, които се съдържат в другия

```
public static List<int> Intersect(  
    List<int> firstList,  
    List<int> secondList)  
{  
    List<int> intersect = new List<int>();  
    foreach (var item in firstList)  
    {  
        if (secondList.Contains(item))  
        {  
            intersect.Add(item);  
        }  
    }  
    return intersect;  
}
```

# Обединение и сечение на списъци – Вариант 1 пример(2)

- Извеждаме списък с метода **PrintList**

```
public static void PrintList(
    List<int> list)
{
    Console.Write("{ ");
    foreach (var item in list)
    {
        Console.Write(item);
        Console.Write(" ");
    }
    Console.WriteLine("}");
}
```

# Обединение и сечение на списъци – Вариант 1

## пример(3)

- В Main() извикваме методите **Add()**, **Union()**, **Interesect()** и **PrintList()**

```
public static void Main()
{
    List<int> firstList = new List<int>();
    firstList.Add(1); firstList.Add(2); firstList.Add(3);firstList.Add(4); firstList.Add(5);
    Console.WriteLine("firstList=");
    PrintList(firstList);
    List<int> secondList = new List<int>();
    secondList.Add(2); secondList.Add(4);
    secondList.Add(6);
    Console.WriteLine("secondList = ");
    PrintList(secondList);
    List<int> unionList = Union(firstList, secondList);
    Console.WriteLine("union = ");
    PrintList(unionList);
    List<int> intersectList = Intersect(firstList, secondList);
    Console.WriteLine("intersect = ");
    PrintList(intersectList);
}
```

# Обединение и сечение на списъци – Вариант 2 с AddRange

- В Main() въвеждаме списъците с метода **Add()**
- извеждаме двата списъка с **PrintList()**

```
public static void Main()
{
    List<int> firstList = new List<int>();
    firstList.Add(1); firstList.Add(2); firstList.Add(3); firstList.Add(4);
    firstList.Add(5); Console.Write("firstList=");
    PrintList(firstList);
    List<int> secondList = new List<int>();
    secondList.Add(2); secondList.Add(4); secondList.Add(6);
    Console.Write("secondList=");
    PrintList(secondList);
    . . .
}
```



# Обединение и сечение на списъци – Вариант 2 с AddRange(1)

- Обединението се конструира така:
  1. Добавяме всички елементи от първия списък
  2. Премахваме тези, които се срещат и във втория
  3. Добавяме елементите на втория с метода AddRange
- извеждаме списъка с **PrintList()**

. . .

```
List<int> unionList = new List<int>();  
unionList.AddRange(firstList);  
for (int i = unionList.Count-1; i >= 0; i--)  
{  
    if (secondList.Contains(unionList[i]))  
    {  
        unionList.RemoveAt(i);  
    }  
}
```

```
unionList.AddRange(secondList);  
Console.Write("union = ");  
PrintList(unionList);
```

. . .

# Обединение и сечение на списъци – Вариант 2 с AddRange(2)

- Конструираме **Сечението** така:
  1. Добавяме елементите на първия с **AddRange**
  2. С **RemoveAt()** премахваме всички, които Не се съдържат и във втория
- извеждаме списъка с **PrintList()**

. . .

```
List<int> intersectList = new List<int>();
intersectList.AddRange(firstList);
for (int i = intersectList.Count-1; i >= 0; i--)
{
    if (!secondList.Contains(intersectList[i]))
    {
        intersectList.RemoveAt(i);
    }
}
Console.Write("intersect = ");
PrintList(intersectList);
}
```

# Алгоритми върху линейни структури от данни. Обединение и сечение на списъци



Въпроси?



# Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство  
на образованието  
и науката



Национална  
програма  
„Обучение за  
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni  
Foundation

