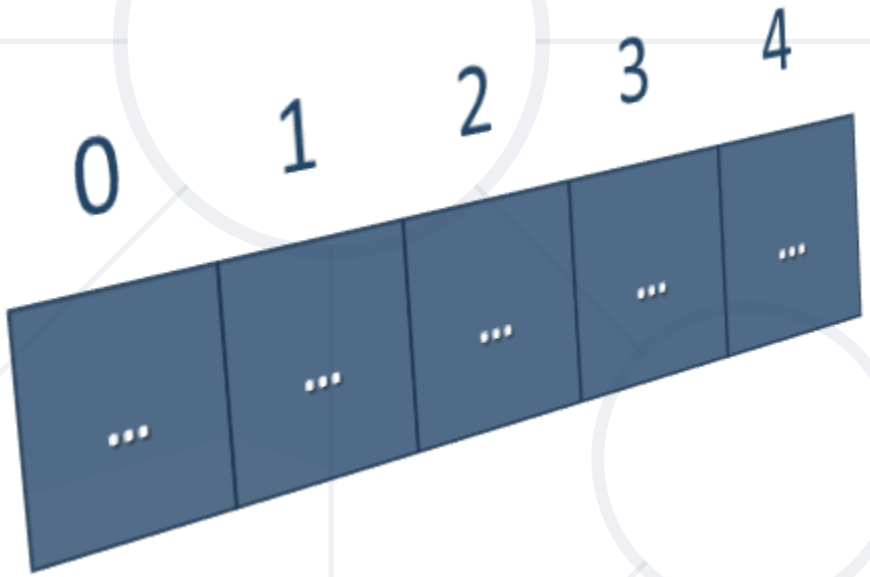
Списък

Обработка на последователни елементи с променлива дължина



СофтУни

Преподавателски екип

Software University

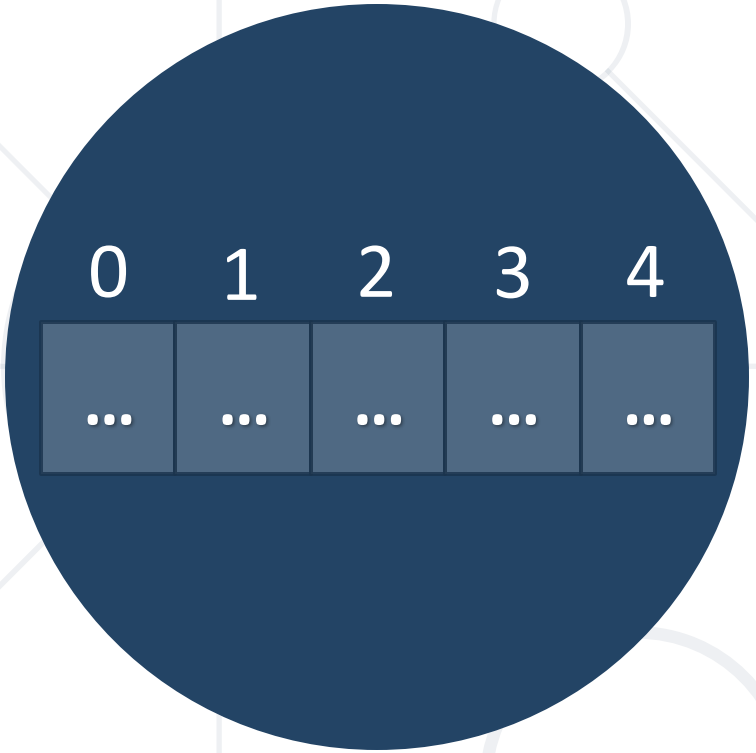
SoftUni

Софтуерен университет

https://softuni.bg

Съдържание

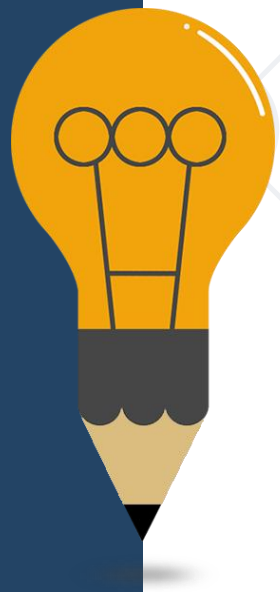
1. Списък
2. Манипулиране на списък
3. Четене на списък от конзолата
4. Сортиране на списък и масив



Списък

Списък от T

- `List<T>` е списък от елементи с еднакъв тип от данни



```
List<string> names = new List<string>();
// Създаване на списък от низове
names.Add("Peter");
names.Add("Maria");
names.Add("George");
// Добавяне на елементи
foreach (var name in names)
  Console.WriteLine(name);
Console.WriteLine(string.Join(", ", names));
// Отпечатване на елементи
```

Списък от T – Основни методи

- Осигуравя операции за **добавяне** / **вмъкване** / **премахване** / **намиране** на елементи:
  - `Add(`елемент`)` – добавя елемент в списъка
  - `Count` – връща броя на елементите в списъка
  - `Remove(`елемент`)` – премахва елемент (връща true / false)
  - `RemoveAt(`индекс`)` – премахва елемент на определен индекс
  - `Insert(`индекс, елемент`)` – добавя елемент на даден индекс
  - `Contains(`елемент`)` – проверява дали елемента съществува в списъка
  - `Sort()` – сортира масива/списъка по азбучен ред

Add() – Добавяне на елемент

- Създаваме празен **лист** и **добавяме** няколко елемента
- Всеки път **броя** на елементите се увеличава

`List<int>` **Брой:** 3

Remove() – Премахване на елемент

- Можем да **премахваме** елемент от **списъка**
- Всеки път **броя** на елементите се намалява

Insert() –Вмъкване на елемент

-10

- **Вмъкваме** елемент на индекс 1
- Индексите на другите елементи се **променят** при вмъкване

List<int>

10

30

Брой: 3

Списък от T – Примери за основни методи

Software University

```
List<int> nums = new List<int> { 10, 20, 30, 40, 50, 60 };
nums.Remove(30);
nums.Add(100);
nums.Insert(0, -100);
Console.WriteLine(string.Join(", ", nums));
Console.WriteLine($"Count: {nums.Count}");
```

```
-100, 10, 20, 40, 50, 60, 100
Count: 7
```

Използване на цикъл или String.Split()

Четене на списък от конзолата

- Първо четем от конзолата **дължината** на списъка:

```
int n = int.Parse(Console.ReadLine());
```

- След това създаваме списък с дължина n и четем елементите:

```
List<int> list = new List<int>();
for (int i = 0; i < n; i++)
{
    int number = int.Parse(Console.ReadLine());
    list.Add(number);
}
// Списъкът се състои от: {10, 20, 30, 40, 50}
```

```
5
10
20
30
40
50
```

Четене на стойностите на списък от един ред

- Списъкът може да бъде прочетен от един ред като стойностите се разделят с интервал:

```
2 8 30 25 40 72 -2 44 56
```

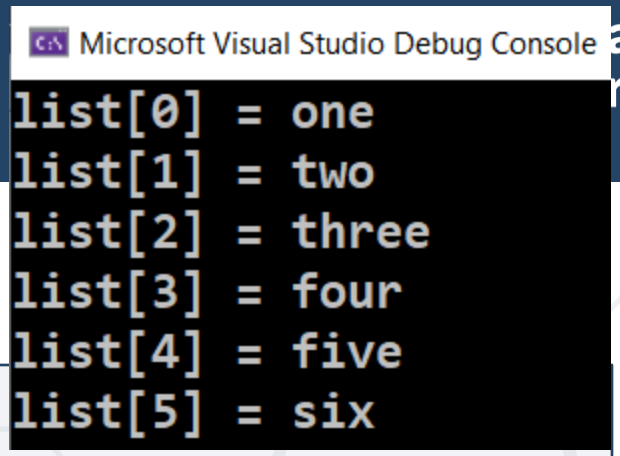
```
string values = Console.ReadLine();
List<string> items = values.Split(' ').ToList();
List<int> nums = new List<int>();
for (int i = 0; i < items.Count; i++)
  nums.Add(int.Parse(items[i]));
```

Превръщане на колекцията в списък

```
List<int> items = Console.ReadLine()
  .Split(' ').Select(int.Parse).ToList();
```

Четене на списък от числа

Принтиране на списъка на конзолата



- Принтиране на списък чрез **for**-цикъл:

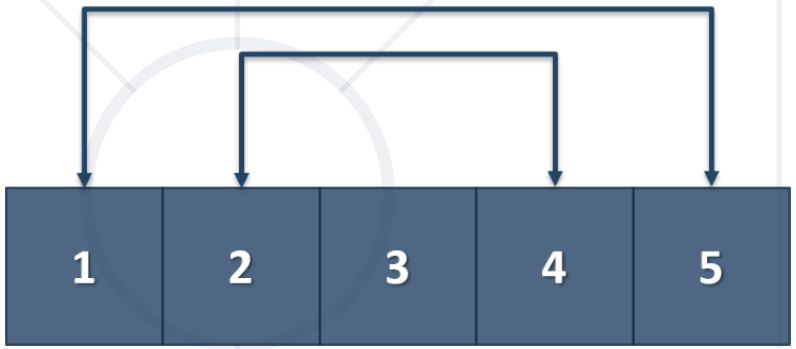
```
List<string> list = new List<string>() {
  "one", "two", "three", "four", "five", "six"};
for (int i = 0; i < list.Count; i++)
  Console.WriteLine("list[{0}] = {1}", i, list[i]);
```

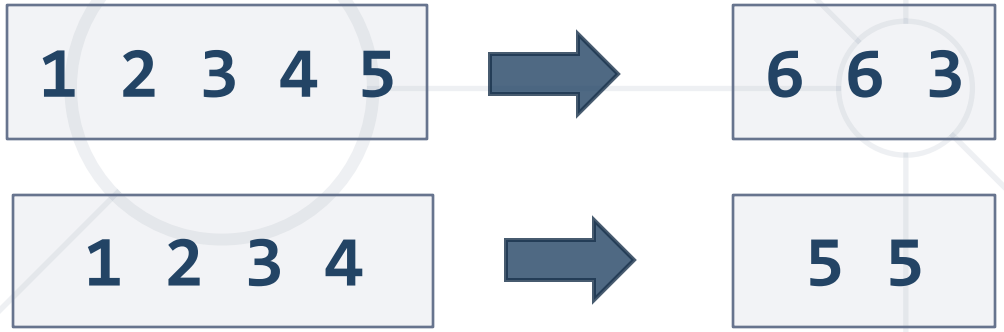
- Принтиране на списък чрез **string.Join(…)**:

```
List<string> list = new List<string>() {
  "one", "two", "three", "four", "five", "six"};
Console.WriteLine(string.Join("; ", list));
// Изход: one; two; three; four; five; six
```

Задача: Трикът на Гаус

- Напишете задача, която **събира всички числа** в **списък** в следният ред:
  - първи + последен, първи + 1 + последен- 1, първи + 2 + последен- 2, … първи + n, последен- n
- Пример:





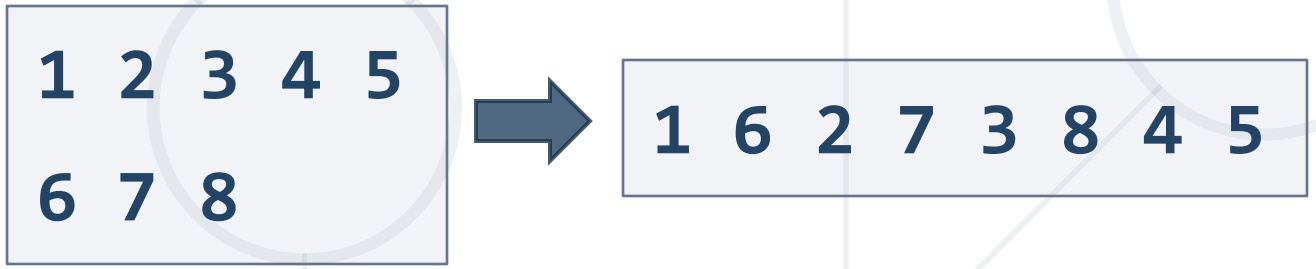
Решение: Трикът на Гаус

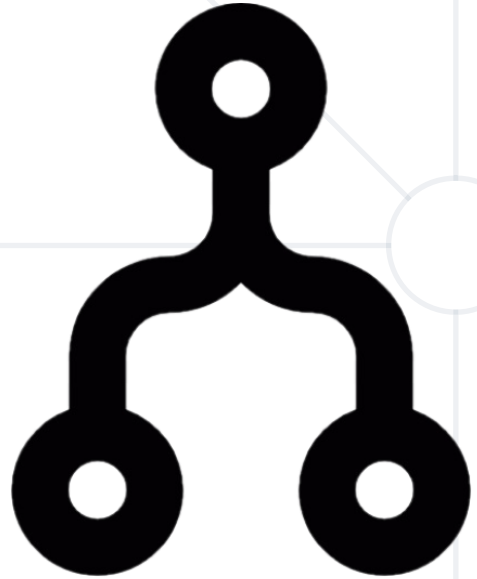
```
List<int> numbers = Console.ReadLine()
   .Split()
   .Select(int.Parse)
   .ToList();
int originalLength = numbers.Count;
for (int i = 0; i < originalLength / 2; i++)
{
   numbers[i] += numbers[numbers.Count - 1];
   numbers.RemoveAt(numbers.Count - 1);
}
Console.WriteLine(string.Join(" ", numbers));
```

Тествайте решението в Judge: https://judge.softuni.org/Contests/Practice/Index/3171#11

Задача: Обединяване на списъци

- Получавате **два списъка с числа**. Принтирайте **изходен списък,** който да съдържа всички цифри от двата списъка.
  - Ако **дължините на двата списъка не са еднакви** просто добавете оставащите елементи в края на списъка
  - list1[0], list2[0], list1[1], list2[1], …





Решение: Обединяване на списъци

```
// TODO: Да се направи вход
List<int> resultNums = new List<int>();
for (int i = 0; i < Math.Min(nums1.Count, nums2.Count); i++)
  // TODO: Да се добавят числата resultNums

if (nums1.Count > nums2.Count)
  resultNums.AddRange(GetRemainingElements(nums1, nums2));
else if (nums2.Count > nums1.Count)
  resultNums.AddRange(GetRemainingElements(nums2, nums1));

Console.WriteLine(string.Join(" ", resultNums));
```

Решение: Обединяване на списъци(2)

```
static List<int> GetRemainingElements(
  List<int> longerList, List<int> shorterList)
{
  List<int> nums = new List<int>();

  for (int i = shorterList.Count; i < longerList.Count; i++)
    nums.Add(longerList[i]);

  return nums;
}
```

Тествайте решението в Judge: https://judge.softuni.org/Contests/Practice/Index/3171#12

Сортиране на списък и масив

Сортиране на списъци

- **Сортиране на списъци** == пренареждане на елементите:**Sort()**
  - Елементите трябва да могат да се **сравняват**, например числа, низове, дати, …

```
List<string> names = new List<string>()
 {"Peter", "Michael", "George", "Victor", "John" };
names.Sort();
Console.WriteLine(string.Join(", ", names));
// George, John, Michael, Peter, Victor
names.Sort();
names.Reverse();
Console.WriteLine(string.Join(", ", names));
// Victor, Peter, Michael, John, George
```

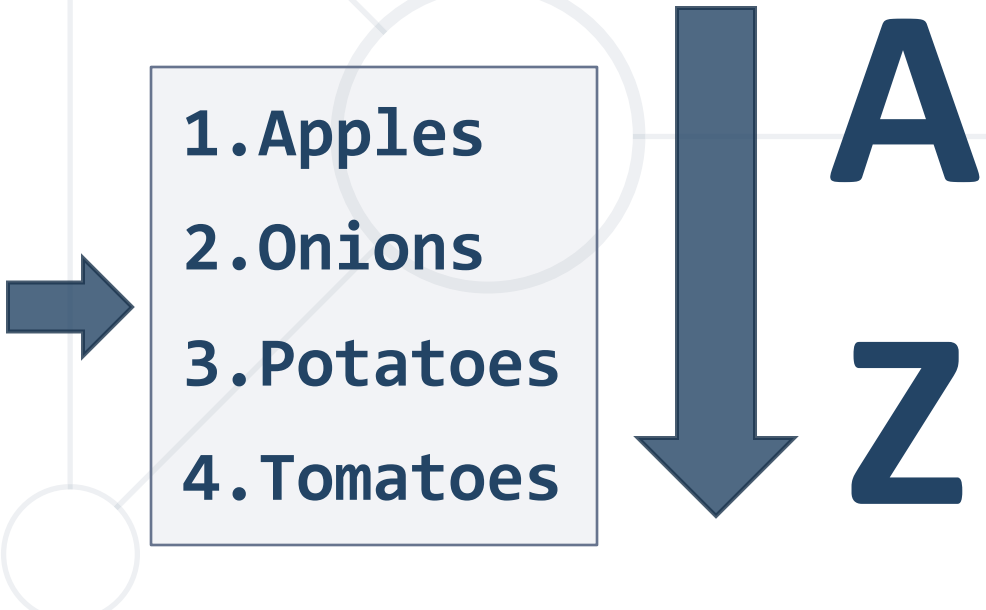
Сортиране във възходящ ред

Обръщане на сортирания списък

Задача: Списък от продукти

- Прочетете числото **n** и след това n на брой редове от **продукти**
  - Принтирайте **номериран списък**, който съдържа всички продукти подредени по име и **по азбучен ред**
- Примери:

```
4
Potatoes
Tomatoes
Onions
Apples
```



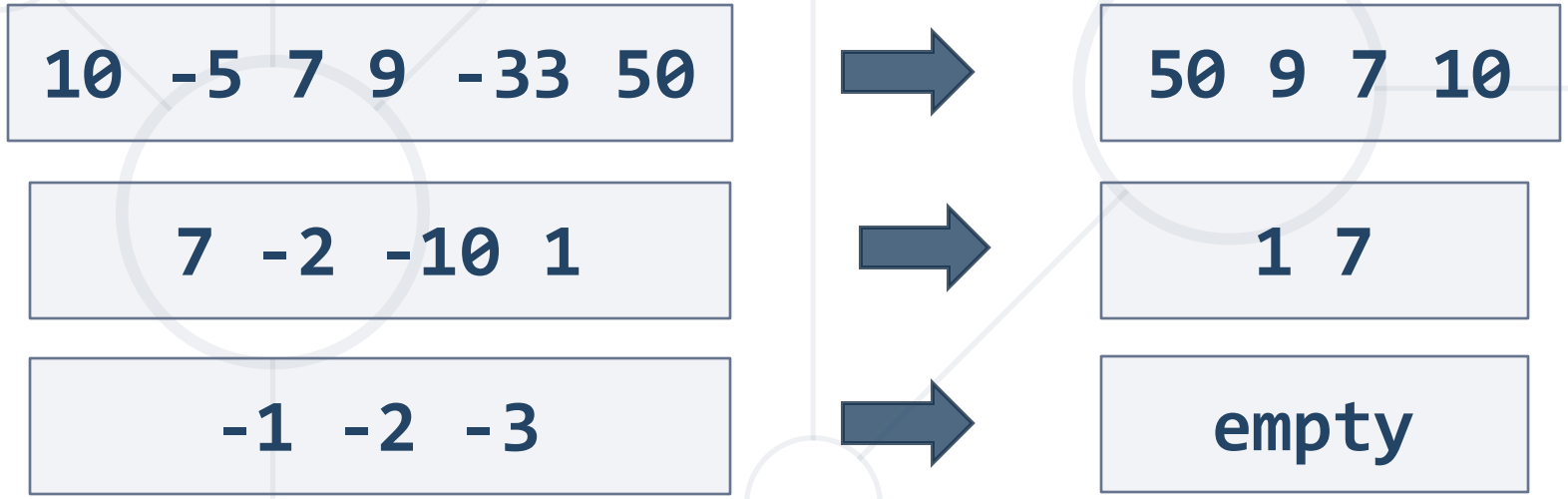
Решение: Списък от продукти

```
int n = int.Parse(Console.ReadLine());
List<string> products = new List<string>();
for (int i = 0; i < n; i++)
{
  string currentProduct = Console.ReadLine();
  products.Add(currentProduct);
}
products.Sort();
for (int i = 0; i < products.Count; i++)
  Console.WriteLine($"{i + 1}.{products[i]}");
```

Тествайте решението в Judge: https://judge.softuni.org/Contests/Practice/Index/3171#13

Задача: Премахнете негативите и обърнете

- Прочетете **списък от числа** и премахнете **всички отрицателни числа**
  - Принтирайте **останалите числа** в **обратен ред**
  - Ако не съдържа числа отпечатайте **"empty"**



Решение: Премахнете негативите и обърнете

```
List<int> nums = // TODO: Да се прочете списъка от конзолата
for (int i = 0; i < nums.Count; i++)
  if (nums[i] < 0) { nums.RemoveAt(i--); }

nums.Reverse();
if (nums.Count == 0)
  Console.WriteLine("empty");
else
  Console.WriteLine(string.Join(" ", nums));
```

Тествайте решението в Judge: https://judge.softuni.org/Contests/Practice/Index/3171#14

Какво научихме днес?

- **Списъците** съдържат редактируема последователност от елементи (с променлива дължина)
- Мога да **добавям** / **премахвам** /**вмъкна** /**модифицирам** елементи по всяко време
- Създаване на списък: `new List<T>()`
- Достъп до елемент чрез индекс: `list[i]`
- Опечатване на елементите на списък: `string.Join(…)`

Въпроси?

SoftUni

Software University | SoftUni Svetlina | SoftUni Creative | SoftUni Digital | SoftUni Foundation | SoftUni Kids

Лиценз

- Този курс (презентации, примери, демонстрационен код, упражнения, домашни, видео и други активи) представлява **защитено авторско съдържание**
- Нерегламентирано копиране, разпространение или използване е незаконно
- © СофтУни – https://softuni.org
- © Софтуерен университет – https://softuni.bg