

# Свързан списък



Учителски екип

Обучение за ИТ кариера

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



# Съдържание

1. Какво са линейните структури данни?
2. АД за списък
3. Статичен списък
4. Свързан списък

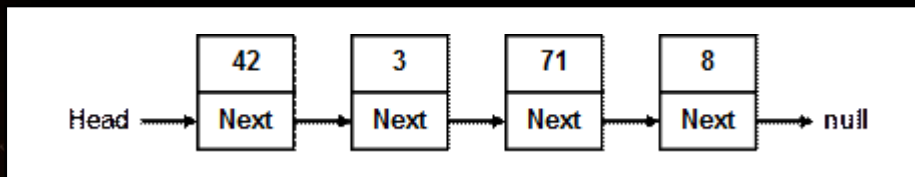


# АТД на Списък

- Списък – **наредена поредица** от елементи, **с променлива дължина**, поддържа следните операции:
  - **Add** (добавяне на елемент)
  - **Get** (извличане на стойност на елемент)
  - **Set** (присвояване на стойност на елемент)
  - **Remove** (премахване на елемент)
- Може да се имплементира чрез масив или списък от свързани възли (**nodes**)

# Свързан списък (1)

- **Свързаният списък** е структура от данни, която съхранява информацията във вид на елементи, в които се пази информация за стойността и за това кой е **следващият елемент**, откъдето идва и името му
- Реализирането на такава структура става чрез създаването на клас **Node**, който описва структурата на един елемент от списъка, а за самият списък се създава клас **DynamicList**





## Свързан списък (2)

- Класът **Node** съдържа следната информация:
  - поле от тип **object**, в което ще се съхранява стойността на елемента
  - поле от **Node**, в което ще се съхранява връзката към следващия елемент

# Реализиране на свързан списък

```
public class Node
{
    private object element;
    private Node next;
    // TODO: Добавете свойства за Element и Next с публични get и set
    public Node(object element, Node prevNode)
    {
        this.element = element;
        prevNode.next = this;
    }
    public Node(object element)
    {
        this.element = element;
        next = null;
    }
}
```

# Реализиране на свързан списък

```
public class DynamicList
{
    // TODO: Дефиниране на Node класа като вложен
    private Node head;
    private Node tail;
    private int count;
    public DynamicList() {...}
    public void Add(object item) { ... }
    public object Remove(int index) { ... }
    public int Remove(object item) { ... }
    public int IndexOf(object item) { ... }
    public bool Contains(object item) { ... }
    public object this[int index] { ...}
}
```

# Реализиране на свързан списък

В конструктора създаваме празен списък

```
public DynamicList()  
{  
    this.head = null;  
    this.tail = null;  
    this.count = 0;  
}
```



# Реализиране на свързан списък

Add:

```
if (head == null)
```

Създаване на първи елемент в списъка

```
else
```

Създаваме пореден елемент в списъка

увеличаваме Count

**tail** и **head** едновременно  
сочат елемента

негов родител е досегашният  
**tail**

# Реализиране на свързан списък

- За част от следващите операции, ще е нужно използването на обхождане на списъка:

Обхождане на списъка:

`index = 0`

Задаваме елемент `current = head`

Докато `current != null`

`current = current.Next`

`index++`

# Реализиране на свързан списък

**Remove (по индекс):**

проверка дали индекса е валиден  
намираме елемента, на съответния индекс  
намаляме Count и изтриваме елемента  
намираме новият последен елемент  
и го задаваме за tail

Чрез обхождане на  
елементите и  
достъпване на Next  
указателя на всеки  
един от тях

# Реализиране на свързан списък

**Remove (по елемент):**

намираме елемента, използвайки обхождане

Ако елементът е намерен:

Намляме Count и го изтриваме

Намираме последният елемент и го задаваме  
на tail

Ако елементът НЕ Е намерен връщаме -1

# Реализиране на свързан списък

**IndexOf:**

намираме елемента, използвайки обхождане

Ако елементът е намерен:

Връщаме индекса му

Ако елементът НЕ Е намерен връщаме -1



# Реализиране на свързан списък

Contains:

Извикваме IndexOf

Ако IndexOf върне стойност различна от -1

връщаме индекса

в противен случай

връщаме -1

# Реализиране на свързан списък

```
public object this[int index]:
```

**get**

проверяваме дали достъпваме валиден индекс  
обхождаме списъка по стандартната схема и  
така достигаме до желания индекс и връщаме  
стойността му

**set**

аналогичен на **get**, с разликата че задаваме  
стойността му, а не я извличаме

# Свързан списък



## 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

