



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

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



Свързан списък

Линейни структури от данни

Съдържание

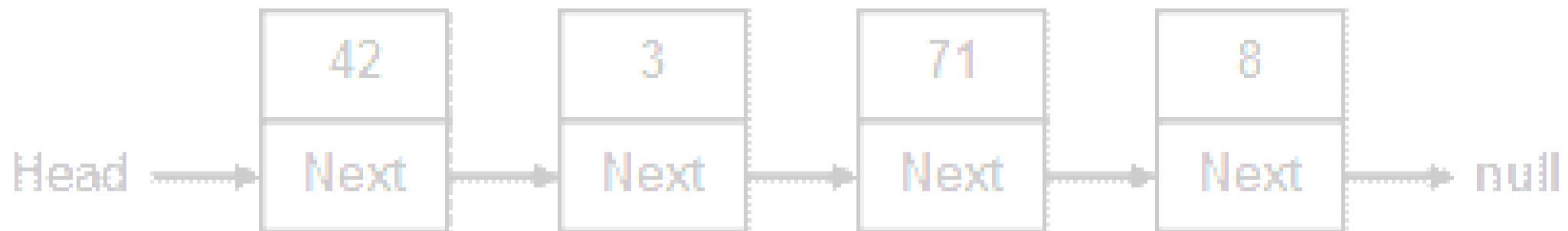
1. Какво са линейните структури данни?
2. Списък
3. Статичен списък
4. Свързан списък

Списък

- **Списък** е 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)

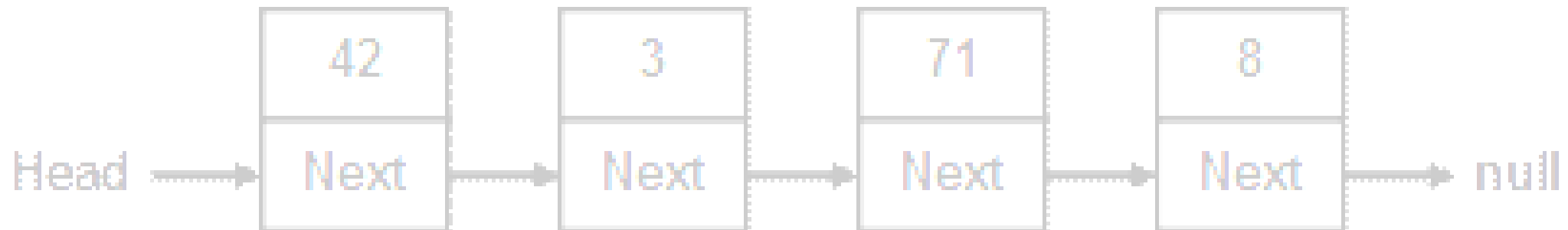
Свързан списък [1/2]

- **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 [2/2]

- 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 [1/10]

```
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 [2/10]

```
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 [3/10]

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

В конструктора
създаваме празен
списък

Реализиране на свързан списък [4/10]

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 [5/10]

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 [6/10]

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

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

Реализиране на свързан списък [7/10]

Remove (по елемент):

намираме елемента, използвайки обхождане

Ако елементът е намерен:

Намаляме Count и го изтриваме

Намираме последният елемент и го задаваме на `tail`

Ако елементът НЕ Е намерен връщаме -1

Реализиране на свързан списък [8/10]

IndexOf:

намираме елемента, използвайки обхождане

Ако елементът е намерен:

Връщаме индекса му

Ако елементът НЕ Е намерен връщаме -1

Реализиране на свързан списък [9/10]

Contains:

Извикваме IndexOf

Ако IndexOf върне стойност различна от -1

връщаме индекса

в противен случай

връщаме -1

Реализиране на свързан списък [10/10]

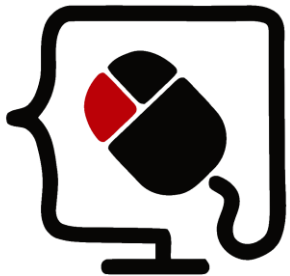
```
public object this[int index]:
```

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



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

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



**SoftUni
Foundation**



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