




ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы

Включил камеру - спас занятие!



Основные коллекции: массив, список, связный список



Нилов Павел
Преподаватель курса

Не забыть включить запись!



Коллекции

Что такое коллекция?

Типы коллекций

Массив

Список

Связный список

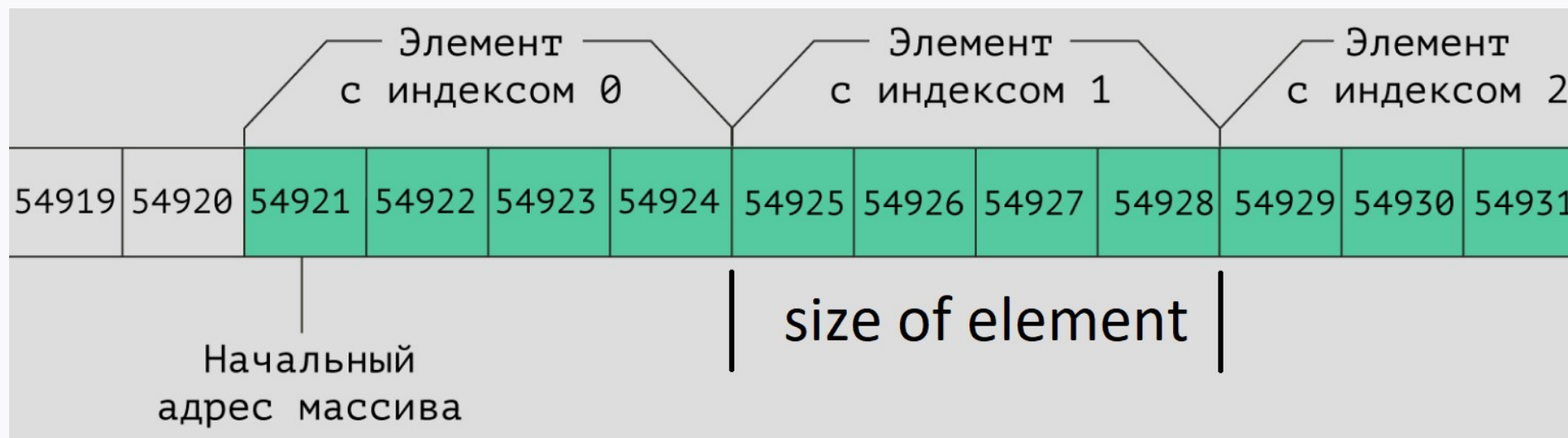
Что такое массив

Массив — это **структура данных** фиксированной длины для хранения упорядоченного набора элементов. Элементы при этом должны относиться к одному типу данных.

Элементы массива физически хранятся в **соседних ячейках памяти**.

К каждому из них можно обратиться по **индексу**.

Элементы массива идут друг за другом в памяти.



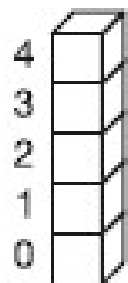
Массив как объект .NET

Особенности массивов в рамках .net:

1. Любой массив - это ссылочный тип данных.
2. Любой массив унаследован от Array.
3. Все массивы реализуют [IList](#). Для получения доступа к элементу по индексу.
4. Все массивы реализуют [IEnumerable](#). Предоставляет перечислитель по коллекции.

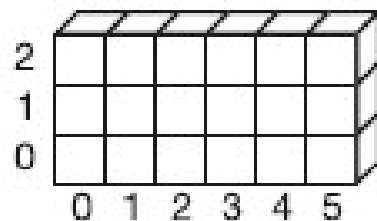
Виды массивов

Одномерный массив

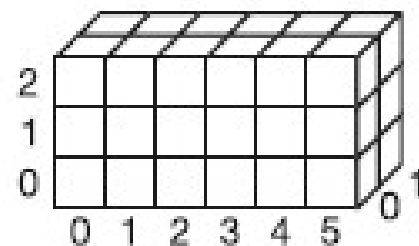


Одномерный массив
`int[5]`

Многомерные массивы

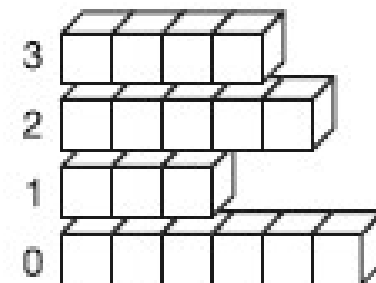


Двухмерный массив
`int[3,6]`



Трёхмерный массив
`int[3,6,2]`

Зубчатый массив



Зубчатый массив
`int[4][]`

Класс Array

Описание класса Array:

<https://learn.microsoft.com/en-us/dotnet/api/system.array?view=net-8.0>

① ② ③ ④ ⑤ ⑥
<T> variable = Array.Method(exemplarOfArray, delegate());

Работа со статическими методами класса Array:

1. Возвращаемый тип: может быть коллекция, квантификатор и пр.
2. Экземпляр переменной.
3. Ключевое слово Array.
4. Статический метод: ForEach(), Sort() etc.
5. Экземпляр массива, над которым будет выполняться операция из п.4.
6. Делегат-обработчик массива.

Вычислительная сложность в массивах

Вычислительная сложность массива:

1. Индексация – $O(1)$. (см. рисунок ниже)

Пояснение: $\text{address} = \text{array}[0] + n * \text{sizeof}(\text{object})$

2. Поиск – $O(n)$. Требуется перебрать все элементы.

3. Вставка – не применимо.

4. Удаление – не применимо.



Вопросы для самопроверки

1. Сколько элементов имеет массив? `int[,] numbers = new int[3, 2, 3]`
2. Сколько измерений (размерность) имеет следующий массив? `int[,] numbers = new int[3, 2, 3]`
3. Что будет выведено на консоль в результате выполнения следующего кода:

Дан следующий массив:

```
int[][] nums = new int [3][];  
nums[0] = new int [2] { 1, 2 };  
nums[1] = new int [3] { 3, 4, 5 };  
nums[2] = new int [5] { 6, 7, 8, 9, 10 };  
Console.WriteLine(nums[3][2]);
```

4. Каким образом мы можем обратиться к числу 7 в массиве выше?

Типы коллекций

Массив

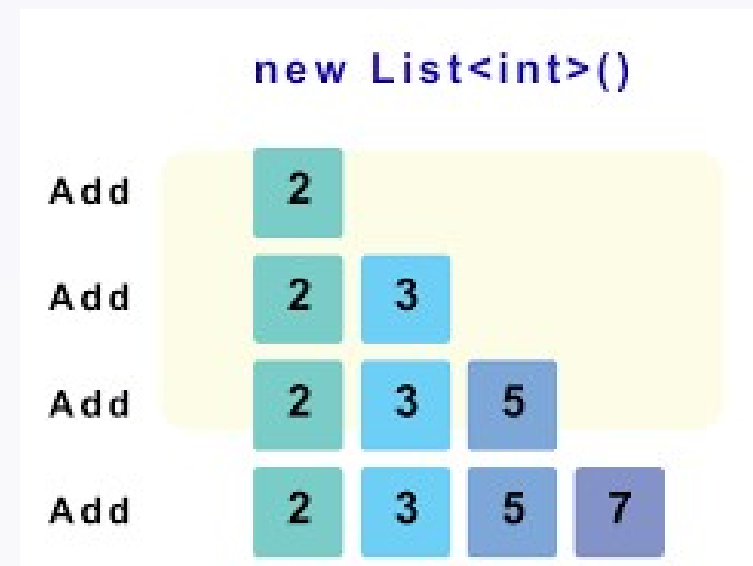
Список

Связный список

Что такое список(List) и как он устроен?

Объект класса `List<T>` представляет строго типизированный список объектов, к которым можно получить доступ по индексу.

1. `List` хранит данные как обычный массив.
2. `List` более удобен в использовании, потому что:
 - i. Нет необходимости заранее задавать размер.
 - ii. Имеет место заранее описанный методов для работы с коллекцией.
 - iii. Добавление элемента в список происходит интуитивно понятным способом.



Добавление элемента

Для добавления нового элемента необходимо сначала выделить новую область памяти в приложения, соответствующей объему массива, а потом скопировать элемент.

[illegible]

Класс List

Описание класса List:

<https://learn.microsoft.com/ru-ru/dotnet/api/system.collections.generic.list-1?view=net-8.0>

Методы класса List:

- Добавление элемента в список
- Поиск элемента в списке
- Вставка элемента в список
- Удаление элемента списка
- Сортировка элементов

Вычислительная сложность в списке

Вычислительная сложность массива:

1. Индексация – $O(1)$.
2. Поиск – $O(n)$.
3. Вставка – $O(n)$.
4. Удаление – $O(n)$.

Типы коллекций

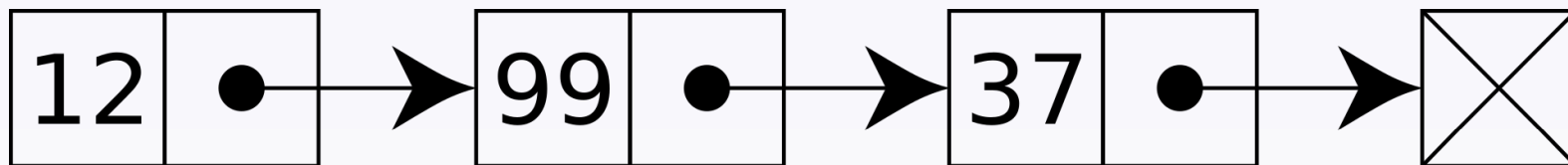
Массив

Список

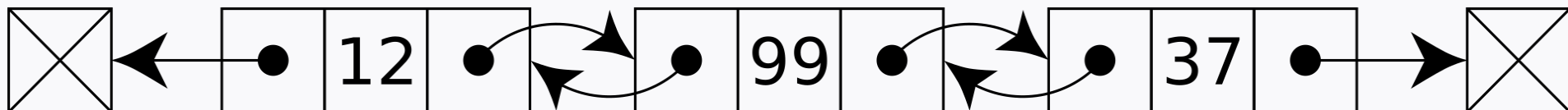
Связный список

Что такое связный список и как он устроен?

Связный список — базовая динамическая структура данных в информатике, состоящая из узлов, содержащих данные и ссылки («связки») на следующий и/или предыдущий узел списка.



В двусвязном списке ссылки в каждом узле указывают на предыдущий и на последующий узел в списке.



Представление данных в памяти приложения

Приведенная в примере с List последовательность будет выглядеть примерно следующим образом.

Это дает преимущество у LinkedList по сравнению с массивами – LinkedList может заполнить практически всю память доступную приложению.

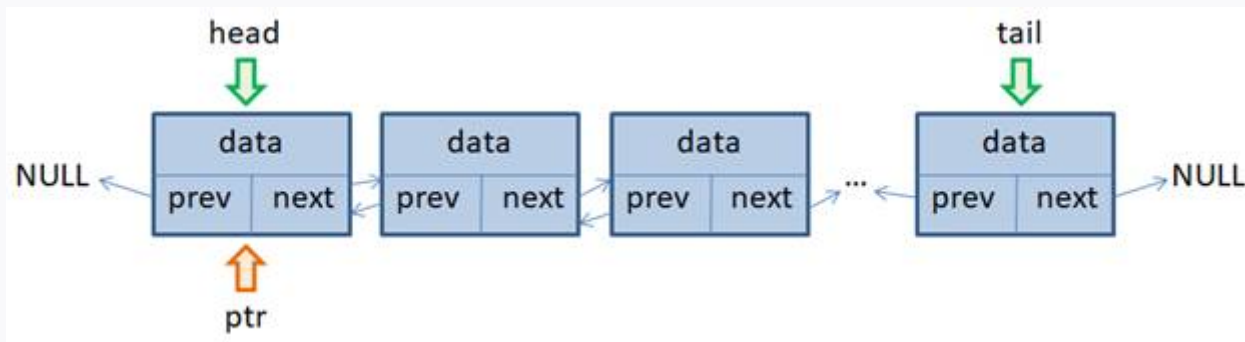
[illegible]

Область применения LinkedList

1. Реализация стеков и очередей.
2. Менеджеры памяти и аллокаторы.
3. Реализация системы отмены действий (undo/redo).

Недостатки:

1. Медленный доступ на чтение.
2. Достаточно простое добавление.
3. Расход памяти на указатели.



Класс LinkedList

Описание класса LinkedList:

<https://learn.microsoft.com/ru-ru/dotnet/api/system.collections.generic.linkedlist-1?view=net-8.0>

Методы класса LinkedList:

- Добавление элемента в связный список
- Вставка элемента в связный список
- Удаление элемента связного списка
- Получение следующего/предыдущего элемента

Вычислительная сложность в связанных списках

Вычислительная сложность массива:

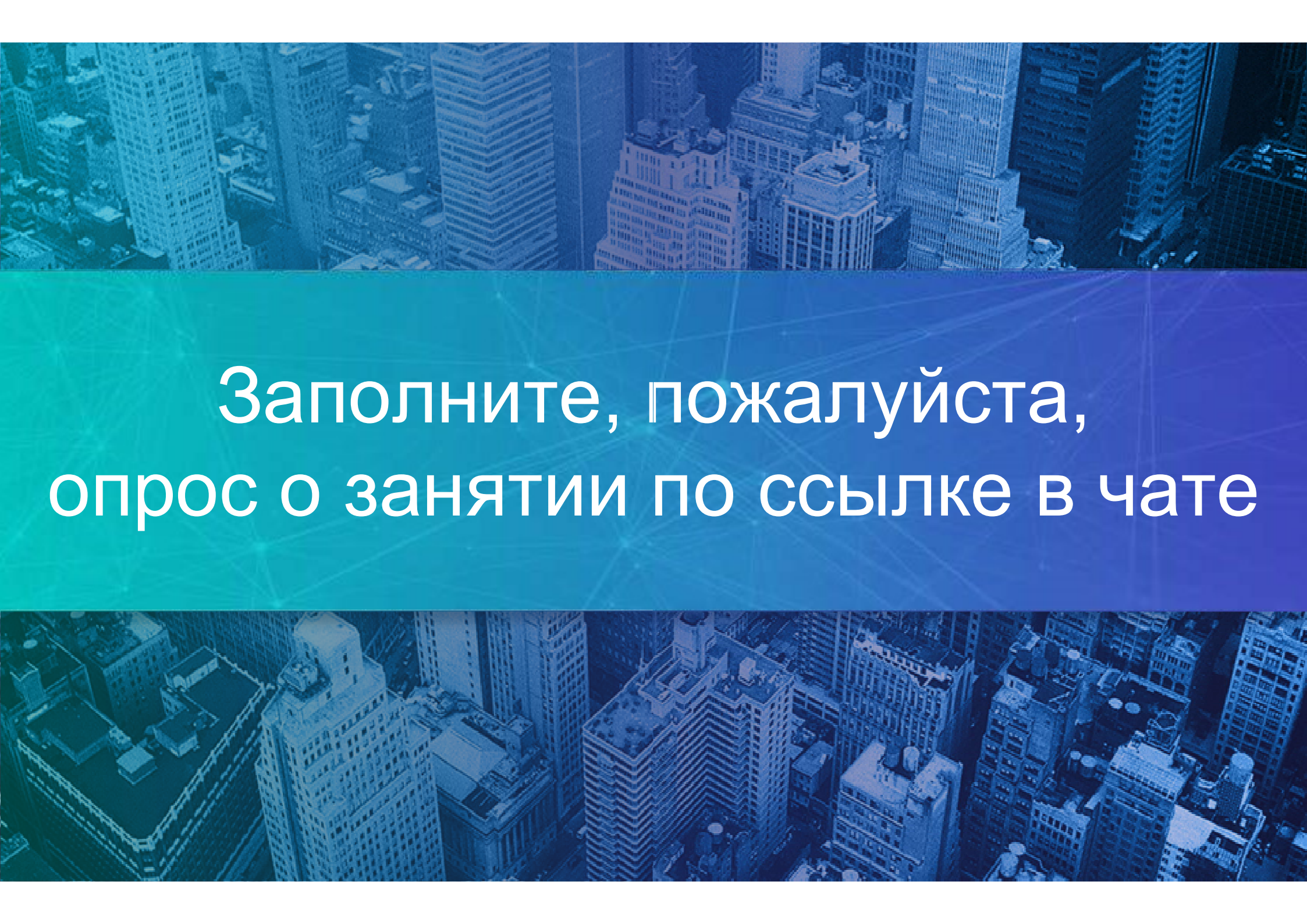
1. Индексация – не применимо.

2. Поиск – $O(n)$.

3. Вставка – $O(1)$.

4. Удаление – $O(1)$.



The background of the slide is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue gradient and a network of white lines connecting various points, suggesting a digital or technological theme.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Ответы на вопросы



Нилов Павел
Преподаватель курса