




ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование

Меня хорошо видно && слышно?

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

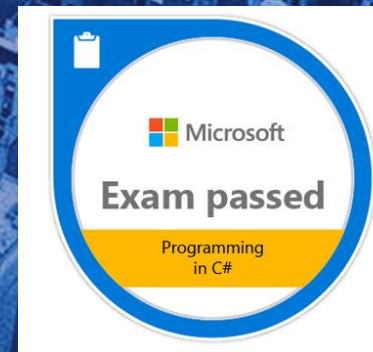
Проверить, идет ли запись!



Основные коллекции: очередь, стек, хешсет, словарь

Приходько Роман

Старший .net разработчик ВІ
к.т.н., доцент СевГУ



Цели

- 1 Понять назначение, строение и особенности коллекций
- 2 Научиться основным операциям с коллекциями
- 3 Разобраться в работе хеш-функции и хеш-таблицы

Маршрут вебинара

Стек и очередь



Хеш-функция и хеш-таблица



Хешсет и словарь

The background of the entire image is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue overlay covers the entire image. Overlaid on this blue background is a network of thin, light blue lines connecting various points, creating a web-like pattern that suggests connectivity or data flow.

Стек и очередь

Происхождение

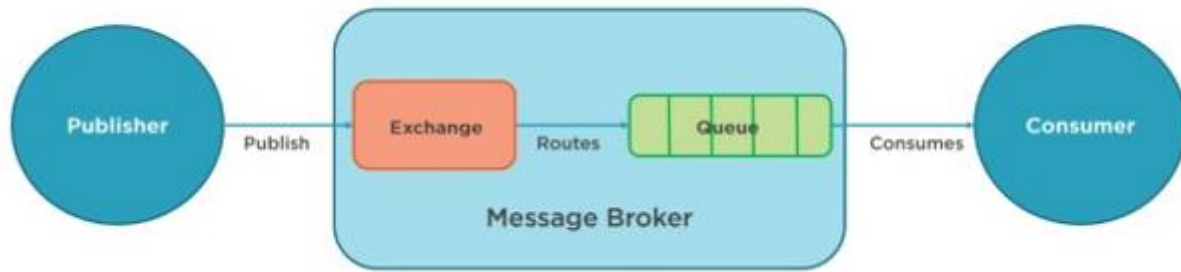


First in, first out
FIFO



First in last out
FILO

Применение



First in, first out
FIFO

```
System.Collections.Generic.KeyNotFoundException: The given key '0' was not present in the dictionary.  
at System.Collections.Generic.Dictionary`2.get_Item(TKey key)  
at Program.Sequence(Int32 start)+MoveNext()  
at Program.Sequence(Int32 start, Int32 end)+MoveNext()  
at Program.MethodAsync()  
at Program.MethodAsync(Int32 v0)  
at Program.MethodAsync(Int32 v0, Int32 v1)  
at Program.MethodAsync(Int32 v0, Int32 v1, Int32 v2)  
at Program.MethodAsync(Int32 v0, Int32 v1, Int32 v2, Int32 v3)  
at Program.Main(String[] args)
```

First in last out
FILO

Назначение

Стек и очередь имитируют физические процессы FILO и FIFO

Основные функции стека

- Push – добавить в стек «сверху»
- Pop – извлечь из стека верхний элемент
- Peek – получить верхний элемент **без извлечения**

Основные функции очереди

- Enqueue – добавить в очередь
- Dequeue – извлечь из очереди
- Peek – получить очередной элемент **без извлечения**

Внутреннее устройство

```
/// <summary>Represents a variable size last-in-first-out (LIFO) collection of instances of the same specified type.</summary>
/// <typeparam name="T">Specifies the type of elements in the stack.</typeparam>
[DebuggerTypeProxy(typeof(StackDebugView<>))]
[DebuggerDisplay("Count = {Count}")]
[TypeForwardedFrom(assemblyFullName: "System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[Serializable]
❖ IL code
public class Stack<T> : IEnumerable<T>, IEnumerable, ICollection, IReadOnlyCollection<T>
{
    #nullable disable
    ❖ IL code
    private T[] _array;
    ❖ IL code
    private int _size;
}
```


Алгоритмическая сложность операций

Добавление элемента – $O(1)$ (если нужно увеличить вложенный массив, то $O(n)$)

Удаление элемента, просмотр элемента – $O(1)$

The image features a high-angle, aerial view of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue and teal gradient. A network of thin, light blue lines connects various points across the image, creating a digital or data network aesthetic. The word "Хешсет" is centered in the middle of the image in a large, white, sans-serif font.

Хешсет

Упорядоченность и неупорядоченность



Array – упорядочена, значит можно перебрать и индексировать
Set - не упорядочен, значит нельзя перебирать и индексировать

Хешсет

Неупорядоченная коллекция **неповторяющихся** элементов

Реализуется классом **HashSet<T>**

Основные методы

Add

Intersect

Union

Except

Алгоритмическая сложность операций

Добавление элемента – $O(1)$ (если нужно увеличить внутренний массив, то $O(n)$)

Удаление элемента, просмотр элемента – $O(1)$

The background of the slide is a blue-tinted aerial photograph of a dense city skyline, likely New York City. Overlaid on this image is a network diagram consisting of numerous small blue dots connected by thin, light blue lines, creating a web-like pattern across the center of the slide.

Хеш-функция. Хеш-таблица

Хеш-функция

Функция, принимающая на вход информацию произвольной длины, и возвращающая информацию фиксированной длины

Это вход хэш-функции.
Вход — это строка бит произвольной длины, которую хэш-функция превращает в строку бит фиксированной длины



→ 1a3fd4128a198fbb3ca345922

Хеш-функция

Достаточно уникальна

Необратима: по хэшу – не понять, что за исходное сообщение

Вычисляется быстро

Хеш-функция

Хранение паролей (достаточно держать хэш пароля, а вводимый пароль – сравнивать по хэшу)

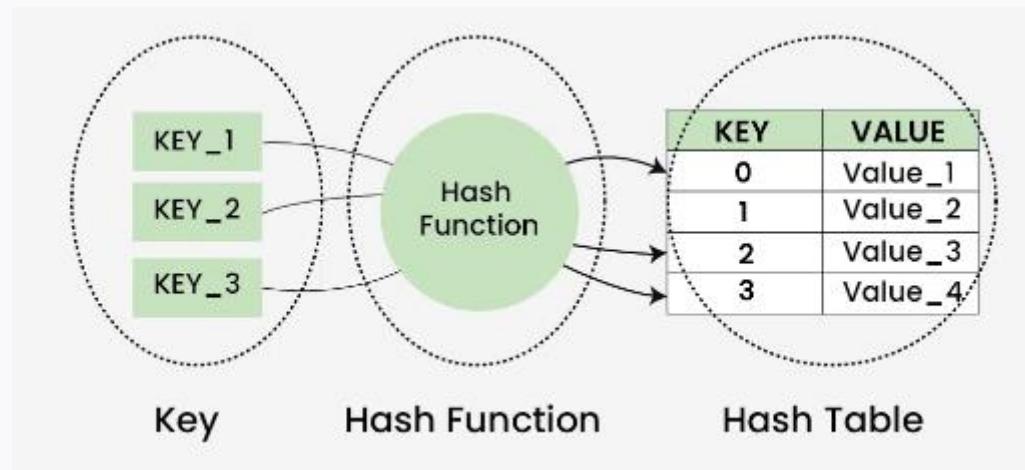
Электронная подпись

Контрольные суммы

Ускорение поиска данных

Хеш-таблица

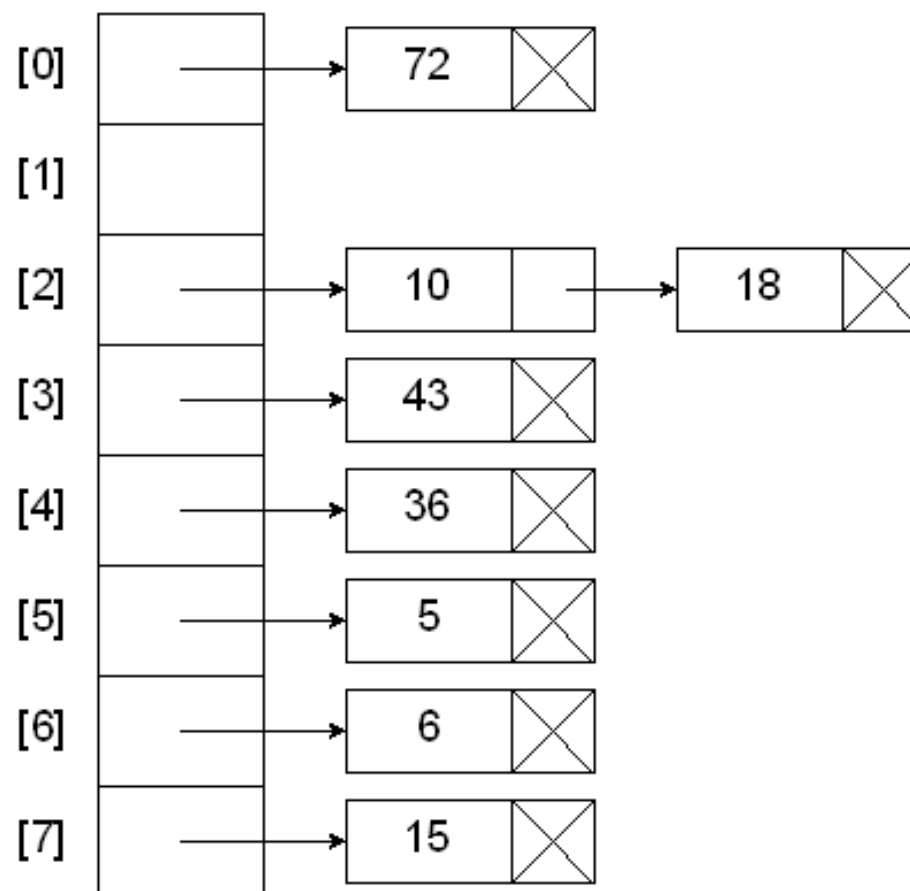
Хеш-таблица (hash table, hash map) – структура данных, использующая хэш-функцию для работы с данными



Хеш-таблица

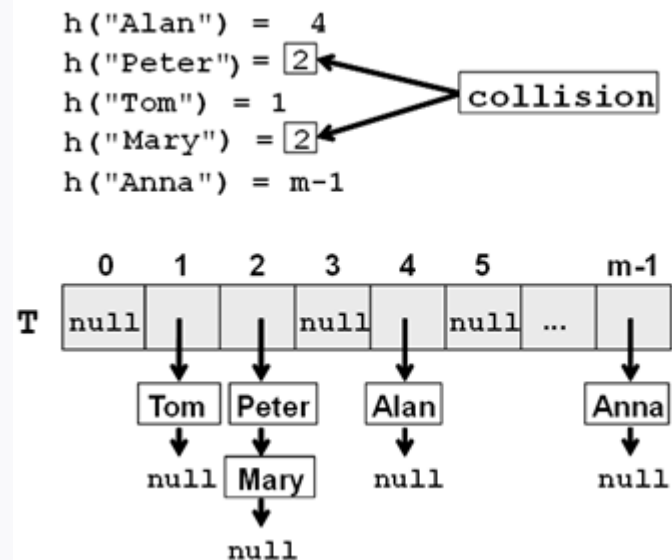
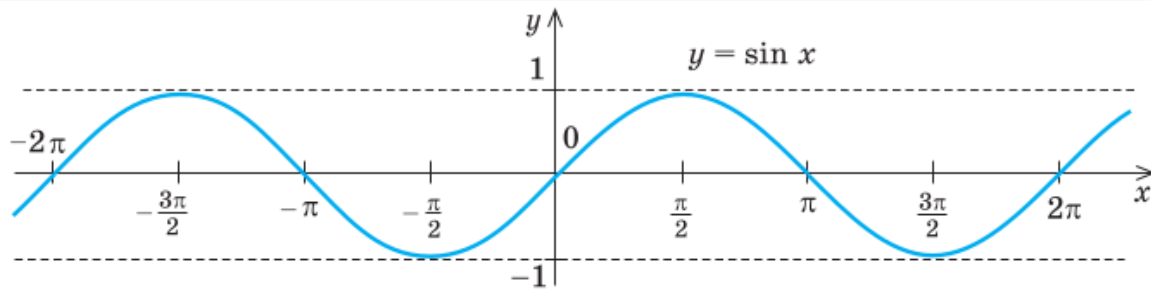
Hash key = key % table size

4	=	36	%	8
2	=	18	%	8
0	=	72	%	8
3	=	43	%	8
6	=	6	%	8
2	=	10	%	8
5	=	5	%	8
7	=	15	%	8



Коллизии

```
var trueHash:int = true.GetHashCode(); trueHash: 1  
var oneHash:int = 1.GetHashCode(); oneHash: 1
```



Bucket

bucket

noun [C]

UK  /'bʌk.ɪt/ US  /'bʌk.ɪt/



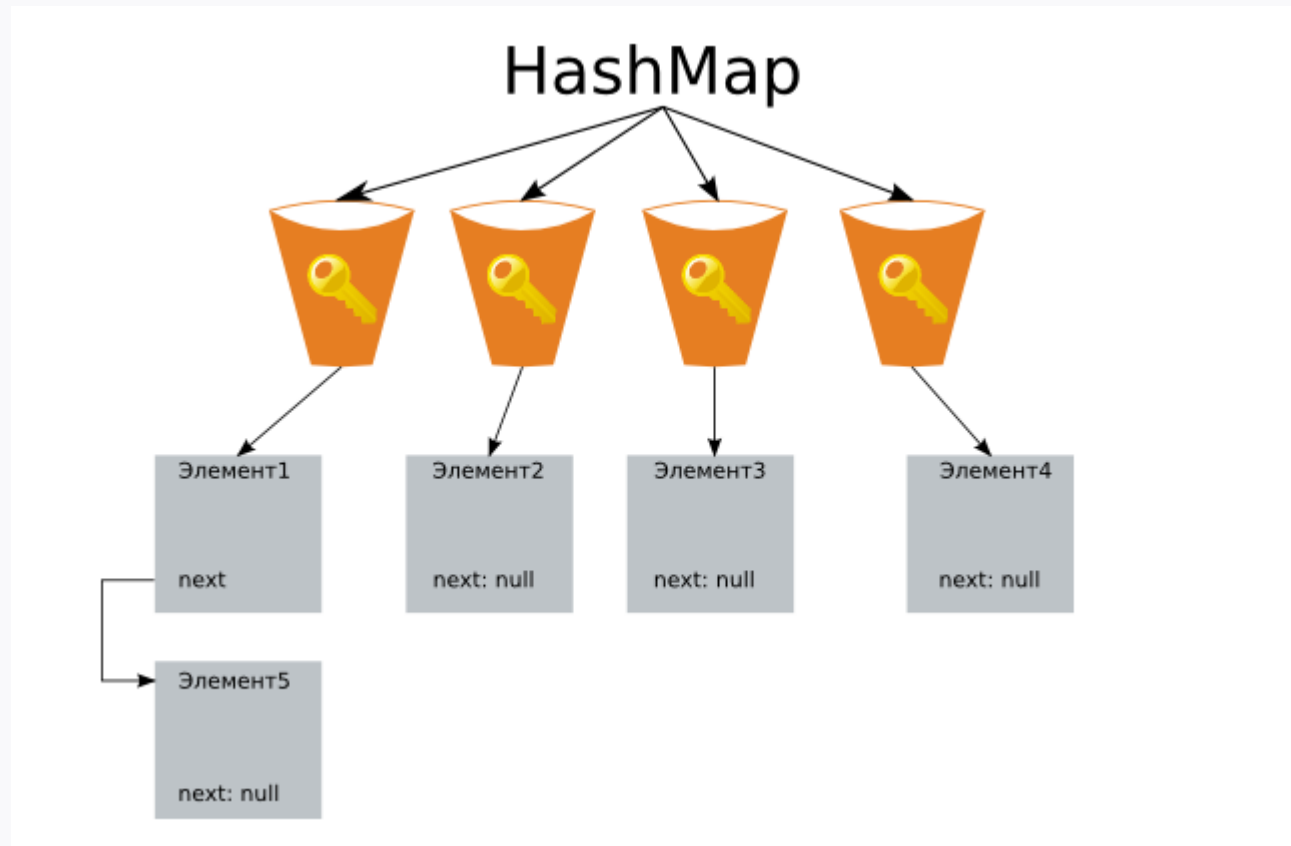
B1

a container with an open top and a handle, often used for carrying liquids:

- *Armed with a bucket and a mop, I started washing the floor.*
- *UK I took my two-year-old nephew down to the beach with his bucket **and spade**.*



Хеш-таблица



Виды адресации

Closed Addressing

Also known as **open hashing**.

0:
1: ①
2: ②②②
3:
4: ④
5:
6:
7: ⑦⑦
8:
9: ⑨

Open Addressing

Also known as **closed hashing**.

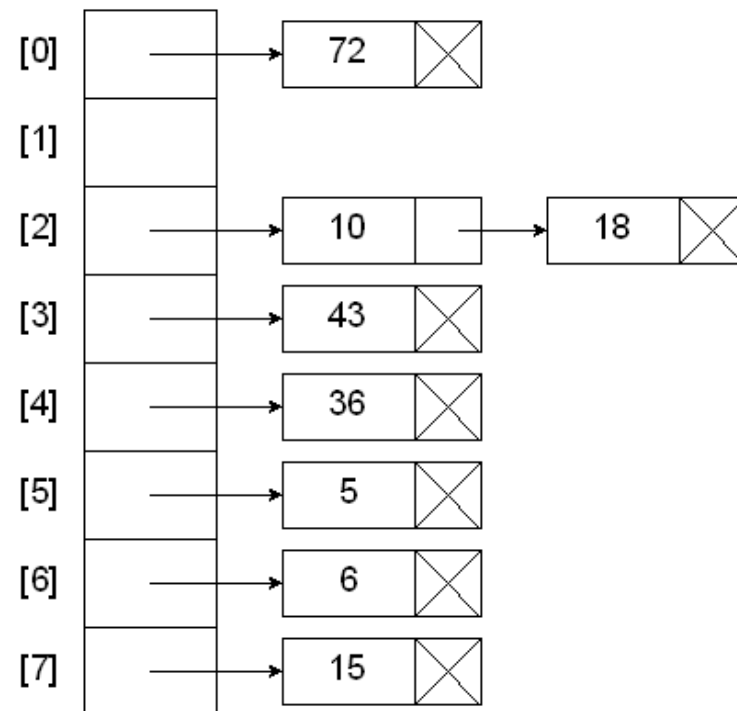
0:
1: ①
2: ②
3: ②
4: ②
5: ④
6:
7: ⑦
8: ⑦
9: ⑨

GetHashCode и Equals

GetHashCode()
Equals(object? obj)

Hash key = key % table size

4	=	36	%	8
2	=	18	%	8
0	=	72	%	8
3	=	43	%	8
6	=	6	%	8
2	=	10	%	8
5	=	5	%	8
7	=	15	%	8



The background of the entire image is an aerial view of a city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue overlay covers the entire image. In the center, there is a network pattern of white lines connecting small dots, creating a web-like structure. The word "Словарь" is written in white, bold, sans-serif font, centered within this network pattern.

Словарь

Словарь

ambition [æmˈbɪʃn] 1) честолюбие 2) стремление	analysis [əˈnæləsɪs] анализ
ambitious [æmˈbɪʃəs] честолюбивый	anarchy [ˈænəki] анархия
ambulance [ˈæmbjuləns] 1) полевой госпиталь 2) машина скорой помощи	anatomy [əˈnætəmi] анатомия
ambush [ˈæmbʊʃ] засада	ancestor [ˈænsɪstə] предок
amends [əˈmendz] <i>pl</i> возмещение; make ~ (for) искупать вину	anchor [ˈæŋkə] якорь
America [əˈmerɪkə] Америка	ancient [ˈeɪnfənt] древний, старинный
American [əˈmerɪkən] 1. <i>a</i> американский 2. <i>n</i> американец	and [ænd] 1) и 2) а, но
amiable [ˈeɪmiəbl̩] любезный, милый	angel [ˈeɪndʒəl] ангел
	anger [ˈæŋɡə] гнев
	angle [æŋɡl̩] <i>mat.</i> угол
	angry [ˈæŋɡri] сердитый; be ~ сердиться
	animal [ˈænɪməl] 1. <i>n</i> животное 2. <i>a</i> животный

Коллекция сопоставления ключа и значения

Реализуется при помощи **Dictionary<TKey, TValue>**

Позволяет реализовать доступ к значению по ключу за постоянное время

Основные методы

Add

["<key>"]

Словарь

Внутри – 2 массива: buckets и entries

buckets – массив с индексами элементов из entries

entries – последовательный массив элементов для хранения

Словарь

Берется `key.GetHashCode()` (hash)

Вычисляется $\text{bucketsIndex} = (\text{hash} \& 0x7fffffff) \% \text{capacity}$, где capacity – размер текущего словаря

В `bucketsIndex` пишется следующий свободный индекс `entries - freeIndex`

В `entries[freeIndex]` пишется значение

Если `entries` закончился – увеличиваем размер и перехешируем

Алгоритмическая сложность операций

Добавление элемента – $O(1)$ (если нужно увеличить внутренний массив, то $O(n)$)

Удаление элемента, просмотр элемента – $O(1)$


Алгоритмическая сложность операций

Добавление элемента – $O(1)$ (если нужно увеличить внутренний массив, то $O(n)$)


Удаление элемента, просмотр элемента – $O(1)$

Вопросы

- Что такое стек?
- Что такое очередь?
- Что такое хэш-функция?
- Перечислите требования к хэш-функции?
- Для чего используется хэшсет?
- Что такое коллизия в словаре и как она решается?

The background of the image is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that features a white network pattern of interconnected dots and lines. The text is centered within this blue layer.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате



Спасибо за внимание!

Приходите на следующие вебинары

Приходько Роман,
Prykhodko.roman@live.com