# Коллекции

*Коллекции* (или совокупности объектов) – описывают стандартные структуры данных. Например: динамический массив, стек, очередь и т.д., и предлагает готовое решение для их использования.

Виды коллекции:

1. **Необобщенные** – хранят элементы как объекты типа object,

Классы описаны в пространстве имен System.Collections

1. **Обобщенный** – в коллекции должны храниться объекты одного типа данных,

Классы описаны в пространстве имен System.Collections.Generic

1. **Параллельные** – представляют эффективные безлопастные операции для доступа к элементам коллекции из нескольких потоков.

Классы описаны в пространстве имен System.Collections.Concorrent

1. Работа с обобщенными коллекциями:
2. Подключить пространство имен

*using System.Collections.Generic*

При объявлении типов коллекции вместо обобщенных типов коллекции пишется конкретный.

Примеры классов обобщенной коллекции:

* List<T> - список предоставляет методы для поиска по списку, его сортировки и изменения
* Dictionary<TKey, TValue> – предоставляет коллекции пар Ключ-Значение (ключи не могут повторяться),
* SortedList<TKey, TValue> - предоставляет коллекцию пар Ключ-Значение упорядоченных по ключу на основе реализации IComparer<T>,
* Stack<T> – это коллекция объектов обрабатываемых по принципу LIFO (последний пришел, первый ушел)
* Queue<T> это коллекция объектов обрабатываемых по принципу FIFO (первый пришел, первый ушел)

1. Работа с коллекцией List:

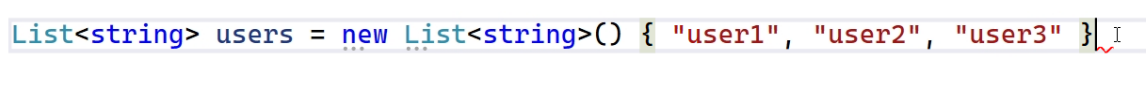
Общая форма:

*List<тип> список = new List<тип>();*

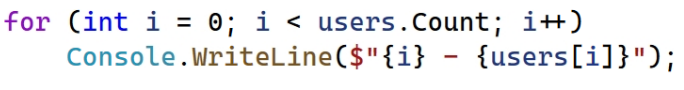
Элементы доступны по индексу.

*список.Count* – это количество элементов в списке

Пример:

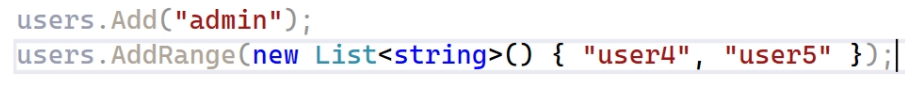


Вывод элементов на экран:

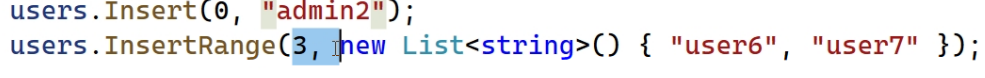


Методы списков:

1. **Добавление элементов в конец списков,**

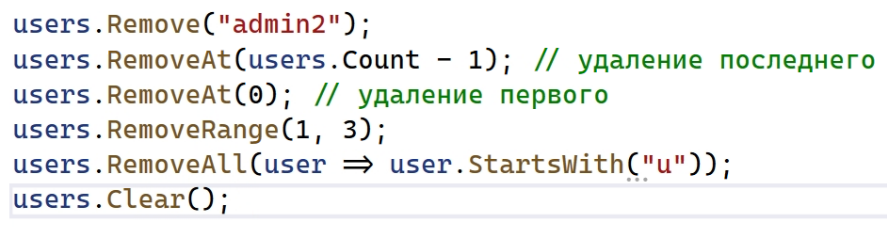


1. **Вставка элементов в список с определённой позиции,**



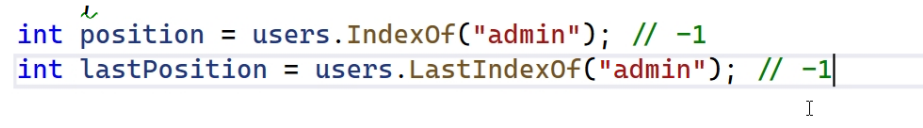
*Первый параметр* — это позиция.

1. **Удаление элементов из коллекции,**

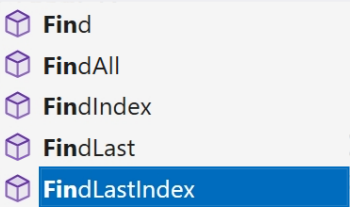


1. **Поиск элементов в списке**

* *if (users.Contains(“…”)) ;* – проверка есть ли элемент в списке
* Проверка для поиска индекса элемента



* *users.Find(предикат)* - метод для поиска по условию(предикату)



* *users.CopyTo(массив)* – копирует массив в список,
* *users.Reverce();* - переворачивает элементы,
* *users.Sort();* - сортирует список.

## Работа c коллекцией Dictionary

Общий вид:

*Dictionary<типКлюча, типЗначения> имяПеременной = new Dictionary<типКлюча, типЗначения>();*

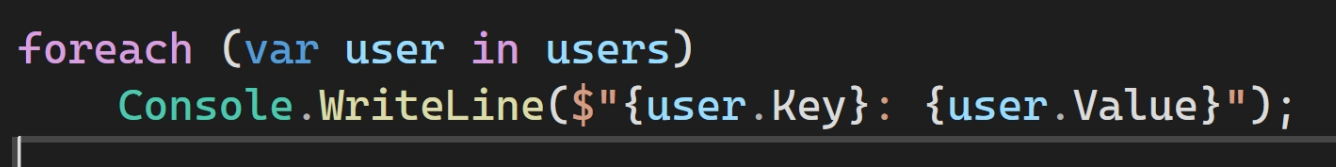
Или

*Dictionary<string, string> users= new(); (тип коллекции Dictionary)*

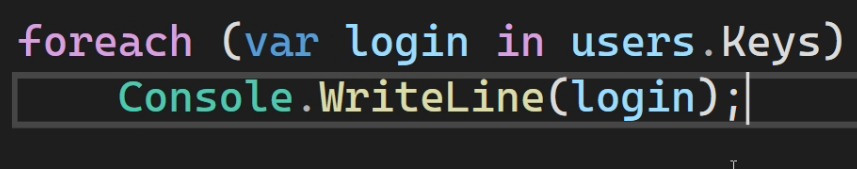
Свойства словаря:

* cловарь.Count // кол-во элементов,
* cловарь.Key // ключ словаря,
* cловарь Value // длина словаря,
* словарь.Add // добавляет объект в конец словаря,
* словарь.TryAdd() // добавляет метод если указанного ключа нет.
* словарь.Remove(); // удаляет объект по ключу
* словарь.Clear(); // очистить словарь

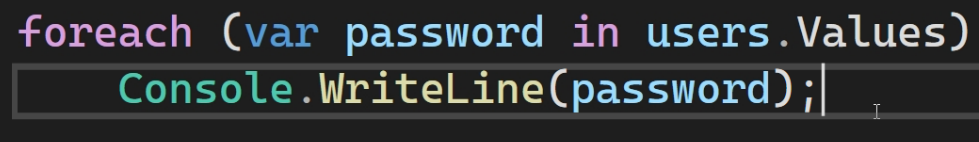
Пример перебора элементов коллекции:



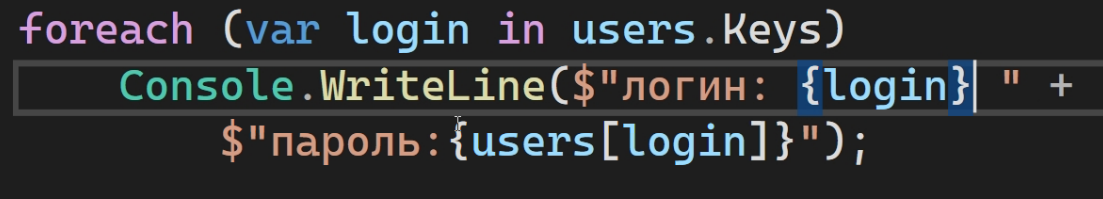
Перебор только ключей



Перебор только длины



Перебор элементов по ключу



Определение есть ли ключ в словаре:

