

Strukture podataka i algoritmi

Kolekcije

dr. sc. Hrvoje Kalinić

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

List

- Dinamički promjenjivo polje (array)
 - Vrsta liste
- Nemaju indekse/ključeve
 - Ne treba se brinuti o veličini


List

```
using System;
using System.Collections.Generic;
class Program
{
    static void Main()
    {
        List<int> list = new List<int>();
        list.Add(2);
        list.Add(1);
        list.Add(3);
        foreach (int broj in list) // foreach petlja
        {
            Console.WriteLine(broj);
        }
        for (int i = 0; i < list.Count; i++) // for petlja
        {
            Console.WriteLine(list[i]);
        }
    }
}
```

List

```
using System;
using System.Collections.Generic;
class Program
{
    static void Main()
    {
        List<int> list = new List<int>();
        list.Add(2);
        list.Add(1);
        list.Add(3);
        foreach (int broj in list) // foreach petlja
        {
            Console.WriteLine(broj);
        }
        for (int i = 0; i < list.Count; i++) // for petlja
        {
            Console.WriteLine(list[i]);
        }
    }
}
```

Korisno za
pretragu



List

```
List<string> imena = new List<string>();  
imena.Add("Ana");  
  
// spoji (konkateniraj) liste (separator: zarez - CSV)  
string line = string.Join(",", imena.ToArray());  
Console.WriteLine(line);  
  
// ubaci element u listu  
list.Insert(pozicija, vrijednost);
```

Kolekcije

- List
- **ArrayList**
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

ArrayList

- Sprema referencu na objekt
 - Objekt može bit različitog tipa
 - Miješana lista
 - List<T> je općenita (generička) klasa
 - "Zastarjela"
 - Savjetuje se korištenje općenite klase List

ArrayList

```
ArrayList bodovi = new ArrayList();
```

```
bodovi.Add(101);
```

```
bodovi.Add(55);
```

```
poz = bodovi.Add(77);
```

```
bodovi.Insert(2, 99);
```

```
bodovi.Insert(2, "maja");
```

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Dictionary

- Pregledavanje/pretraživanje po ključu (riječi, indeksu)
 - Riječnik

Dictionary

```
class Program
{
    static void Main()
    {
        Dictionary<string, int> dictionary = new Dictionary<string,
int>();
        dictionary.Add("pas", 5);
        dictionary.Add("miš", 2);
        dictionary.Add("muha", 0);
        dictionary.Add("ameba", -10);

        // imamo li vrijednost u riječniku
        if (dictionary.ContainsKey("muha"))
        {
            int value = dictionary["muha"];
            Console.WriteLine(value);
        }
    }
}
```

Dictionary

```
// petlja po parovima
foreach (KeyValuePair<string, int> pair in d)
{
    Console.WriteLine("{0}, {1}", pair.Key, pair.Value);
}
// korištenje ključa riječi var za pobrojavanje (enumerate) riječika
foreach (var pair in d)
{
    Console.WriteLine("{0}, {1}", pair.Key, pair.Value);
}
```

Kolekcije

- List
- ArrayList
- Dictionary
- **Hashtable**
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Hashtable

- Asocijativno polje
- Optimizacija pronalaženja (lookups)
 - Svaki ključ dobiva svoj hash (asocijaciju, "šifru")
- Stariji i sporiji od rječnika

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- **Tuple**
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Tuple

- "Nepromjenjiva polja"

```
// Create a 7-tuple.
var population = new Tuple<string, int, int, int, int, int, int>(
    "New York", 7891957, 7781984,
    7894862, 7071639, 7322564, 8008278);
// Display the first and last elements.
Console.WriteLine("Population of {0} in 2000: {1:N0}",
    population.Item1, population.Item7);
// The example displays the following output:
//
```

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Stog

- Eng. Stack
- LIFO – eng. Last-in-first-out
- Općenita klasa

- Metode
 - Push(), Pop(), Peek()
 - Contains()
 - Clear(), Count()

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Red

- Eng. Queue
- FIFO – eng. First-in-first-out
- Općenita klasa

- Metode
 - Enqueue(), Dequeue(), Peek()
 - Contains(), Clear(), ToArray(), Count()
 - TrimExcess()

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Skup

- Eng. Set
 - HashSet
 - SortedSet

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- DoubleLinkedLists
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Povezana lista (jednosmjerna)

- Eng. Linked list
- Brzo ubacivanje i izbacivanje
 - pokazivači
- Metode
 - AddLast(), AddFirst()
 - AddAfter(), AddBefore()
 - Find(), Remove()

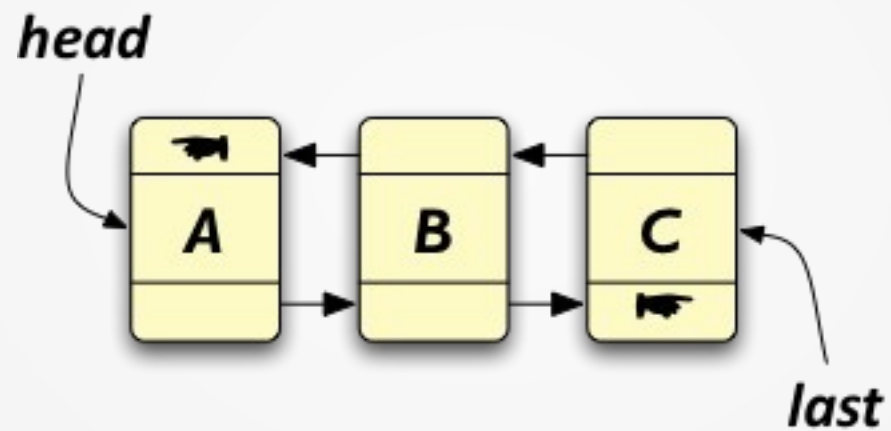
Povezana lista (jednosmjerna)

```
class Program
{
    static void Main()
    {
        // nova lista i dodavanje elemenata
        LinkedList<string> linked = new LinkedList<string>();
        linked.AddLast("pas");
        linked.AddLast("svi");
        linked.AddLast("oni");
        linked.AddFirst("prvi");
        // dodaj cvor nakon prvog
        LinkedListNode<string> node = linked.Find("prvi");
        linked.AddAfter(node, "dodatan");
        // ispis
        foreach (var item in linked)
        {
            Console.WriteLine(item);
        }
    }
}
```

Kolekcije

- List
- ArrayList
- Dictionary
- Hashtable
- Tuple
- Stack
- Queue
- Sets
- LinkedLists
- **DoubleLinkedLists**
- KeyValuePair
- Concurrent
- MultiMap
- * Dictionary *
- ...

Dvosmjerna povezana lista



Kolekcije - napomena

- `CollectionBase`
 - Podržana (supported)
 - Samo-kompatibilnost (backward compatible)
 - Zastarjela (obsolete)
- Prednost dati kolekcijama iz (namespace-a)
 - `System.Collections.Generic`
 - Ne `System.Collections.ObjectModel`

Kolekcije - napomena

- Prednosti
 - Brzina
 - Mogu (nasljeđivanjem) stvoriti kolekcije fiksnog tipa (strongly-typed)
- Česte negeneričke kolekcije
 - CollectionBase
 - ArrayList
 - HashTable