Linked List Implementation in C#

chandrasen singh

6 Jun 2016 CPOL

Linked list implementation in C#

CodeProjectIn this article, I'll explain about linked list, the pros and cons of using linked list and then implementation of linked list in C#. However, Microsoft provides strongly typed linked list `LinkedList(T)` in `System.Collections.Generic` namespace. Here, I'll explain the core implementation of Linked List.

What is Linked List?

Linked list is a linear data structure. It's a collection of elements. Element is called as `Node`.
Each element has value(data) and reference of next node. The very first node is called as `Head` and last element has reference to `null` value.

Types of Linked Lists

Basically, there are 3 types of linked lists.

1. **Singly Linked List**: A singly linked list is a list which has a value(data) and a reference to next node. In this, last node's next reference will be `null`.
2. **Doubly Linked List**: A doubly linked list is a list which has a data and 2 references, one for next node and another for previous node and last node's next reference will be `null`.
3. **Circular Linked List**: In circular linked list, last node's next reference will be head or first element. Circular linked list can be singly linked list or doubly linked list.

In this article, I'll explain implementation of only singly linked list.

Pros and Cons

Main advantage of linked list is that it is a dynamic data structure. Unlike in array where length of array is predefined, we can add dynamic number of data in linked list without worrying about size of linked list. So linked list can be used where there is an unknown number of data that needs to be stored at run time.

Disadvantage of linked list is that it takes more space than array. Because it stores reference of next/previous nodes.

Implementation of Linked List

Node Class

Here is the node class which has 2 properties:

C# Copy Code

```
public class Node
{
    public Node Next;
    public object Value;
}
```

One property is '`Next`', which will have reference of next node and another is '`Value`', which will have data in this node.

LinkedList Class

Now implement linked list class as follows:

C# Copy Code

```
public class LinkedList
{
   private Node head;
   private Node current;//This will have latest node
   public int Count;
}
```

'`head`' will have head node. '`current`' will have latest node, i.e., tail node and '`Count`' will return total number of nodes in linked list.

Constructor

Initially, `head` and `current` value will be the same and will have empty node, so we will create constructor for that setting.

C# Copy Code

```
public LinkedList()
{
  head = new Node();
  current = head;
}
```

Operations in Linked List

Now, we will create some operations on linked list.

1. Add node after last element:

   C# Copy Code

   ```
   public void AddAtLast(object data)
   {
      Node newNode = new Node();
      newNode.Value = data;
      current.Next = newNode;
      current = newNode;
      Count++;
   }
   ```

   This method will add node after tail node and it will increase count by one. Similarly, you can add node as first node.

2. Add node as fist element:

C# Copy Code

```
public void AddAtStart(object data)
{
  Node newNode = new Node() { Value = data};
  newNode.Next = head.Next;//new node will have reference of head's next reference
  head.Next = newNode;//and now head will refer to new node
  Count++;
}
```

3. Remove node from start:

C# Copy Code

```
public void RemoveFromStart()
{
   if (Count > 0)
   {
     head.Next = head.Next.Next;
     Count--;
   }
   else
   {
     Console.WriteLine("No element exist in this linked list.");
   }
}
```

Similarly you can write method to remove node from last or from any index. You have to do traverse from head to that particular index and remove node as above by changing the reference.

4. Traverse whole linked list:

C# Copy Code

```
/// <summary>
/// Traverse from head and print all nodes value
/// </summary>

public void PrintAllNodes()
{
//Traverse from head
Console.Write("Head ->");
Node curr = head;
while (curr.Next != null)
{
curr = curr.Next;
Console.Write(curr.Value);
Console.Write("->");
}
Console.Write("NULL");
}
```

Start from head and traverse until you get next node as `null`.

Note: Similarly, you can write code for deleting node of specific index, inserting node on specific position, etc.

Time to Celebrate...

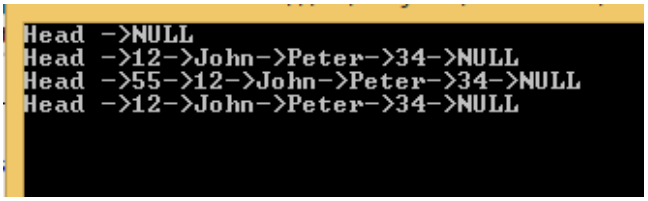
Now it's time to test our code. To test our code, we will write `main` method and call linked list and its operations in that.

C# Copy Code

```
class Program
{
    static void Main(string[] args)
```

```
    {
        LinkedList lnklist = new LinkedList();
        lnklist.PrintAllNodes();
        Console.WriteLine();

        lnklist.AddAtLast(12);
        lnklist.AddAtLast("John");
        lnklist.AddAtLast("Peter");
        lnklist.AddAtLast(34);
        lnklist.PrintAllNodes();
        Console.WriteLine();

        lnklist.AddAtStart(55);
        lnklist.PrintAllNodes();
        Console.WriteLine();

        lnklist.RemoveFromStart();
        lnklist.PrintAllNodes();

        Console.ReadKey();
    }
}
```

Guess what the output will be. Here it is.



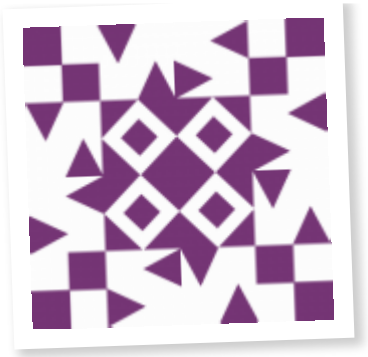
Note: The post first appeared on www.w3techno.blogspot.com.
You can **download** this project. To download, **CLICK HERE.**

Did you like this article? Feel free to reach me by leaving comments below.

License

Share

About the Author



chandrasen singh

Software Developer (Senior)
India

Hi I love code and explore coding, thing at machine level.I have 6 years of experience in .net technologies. I have worked on c#,asp.net MVC,EF,Linq,AngularJs,Node js.I do blog at www.w3techno.blogspot.com . When I do't do coding I love to travel and read books.

Comments and Discussions

You must Sign In to use this message board.

Search Comments

First Prev Next

Why using 'head.next' during AddAtStart method
Member 13929090 30-May-21 12:16

How to add removeatlast on this program?
Member 14143547 7-Feb-19 13:52

How the Head is updated every time we call the AddLast() function?
Pradeep Kumar Kalsyan 22-Nov-17 3:34

My vote of 5!
jediYL 9-Jun-16 18:12

Why are the methods not static?
rhyous 8-Jun-16 10:46

Re: Why are the methods not static?
Markus64 21-Jun-16 0:42

Re: Why are the methods not static?
rhyous 22-Jun-16 9:53

More differences between a Linked List and an Array
Michael Epner 8-Jun-16 1:29

Re: More differences between a Linked List and an Array
Ehsan Sajjad 19-Mar-17 22:09

Why not generic ?
webmaster442 7-Jun-16 19:59

My vote of 3

Muhammad Shahid Farooq 6-Jun-16 21:35

Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.