|  |
| --- |
| public class DynamicList |
|  |  |
|  | // Списък |
|  | private Node head; |
|  | private Node tail; |
|  | private int count; |
|  | public int Count // O(1) |
|  | { |
|  | get { return count; } |
|  | set { count = value; } |
|  | } |
|  |  |
|  | // Конструктор |
|  | public DynamicList() |
|  | { |
|  | this.head = null; |
|  | this.tail = null; |
|  | this.count = 0; |
|  | } |
|  |  |
|  | // Добавяне на елементи |
|  | public void Add(object item) |
|  | { |
|  | if (count == 0) |
|  | { |
|  | Node current = new Node(item); // Първи |
|  | head = current; |
|  | tail = current; |
|  | } |
|  | else |
|  | { |
|  | Node current = new Node(item, tail); // Следащ |
|  | tail = current; |
|  | } |
|  | count++; |
|  | } |
|  |  |
|  | // Индекса на елемент |
|  | public int IndexOf(object item) |
|  | { |
|  | Node current = head; |
|  | int index = 0; |
|  | while (current != null) |
|  | { |
|  | if (current.Element.Equals(item)) return index; |
|  | index++; |
|  | current = current.Next; |
|  | } |
|  | return -1; // Not Found |
|  | } |
|  |  |
|  | // Премахване |
|  | public object Remove(int index) |
|  | { |
|  | int currentIndex = 0; |
|  | Node current = head; |
|  | Node previous = null; |
|  | while (current != null) |
|  | { |
|  | if (currentIndex == index) |
|  | { |
|  | if (previous != null) previous.Next = current.Next; |
|  | else this.head = current.Next; |
|  | if (current.Next == null) this.tail = previous; |
|  | count--; |
|  | return current.Element; |
|  | } |
|  | else |
|  | { |
|  | previous = current; |
|  | current = current.Next; |
|  | currentIndex++; |
|  | } |
|  | } |
|  | return null; |
|  | } |
|  | public int Remove(object element) |
|  | { |
|  | int index = this.IndexOf(element); |
|  | object removed = this.Remove(index); |
|  | if (removed == null) return -1; |
|  | else return index; |
|  | } |
|  |  |
|  | // Търсене |
|  | public bool Contains(object item) |
|  | { |
|  | if (IndexOf(item) != -1) return true; |
|  | else return false; |
|  | } |
|  |  |
|  | // Итератор |
|  | public object this[int index] |
|  | { |
|  | get |
|  | { |
|  | int currentIndex = 0; |
|  | Node current = head; |
|  | Node returnNode = null; |
|  | while (current != null) |
|  | { |
|  | if (currentIndex == index) |
|  | { |
|  | returnNode = current; |
|  | break; |
|  | } |
|  | current = current.Next; |
|  | currentIndex++; |
|  | } |
|  | if (returnNode != null) return returnNode.Element; |
|  | throw new IndexOutOfRangeException(); |
|  | } |
|  | set |
|  | { |
|  | int currentIndex = 0; |
|  | Node current = head; |
|  | Node changedNode = null; |
|  | while (current != null) |
|  | { |
|  | if (currentIndex == index) |
|  | { |
|  | changedNode = current; |
|  | break; |
|  | } |
|  | current = current.Next; |
|  | currentIndex++; |
|  | } |
|  | if (changedNode != null) changedNode.Element = value; |
|  | throw new IndexOutOfRangeException(); |
|  | } |
|  | } |