

Telerik Software Academy

Technical Support Course - Final Exam (August 12, 2013)

Task 1 - Ticket

Hello Telerik Support,

I'm trying to build a double-ended queue ("dequeue") using **LinkedList<T>** (from the namespace **System.Collections.Generic**). Would you please help me to better understand and build it?

My requirement is that the Dequeue must also inherit **IDeque<T>** interface and implement all of its methods and properties.

Is it also possible that my implementation does not place fixed limits on the number of elements it may contain?

Would you please help me to better understand how I can build such a queue? It would be very helpful you could also share what are your suggestions and recommendations on the matter?

I will be very grateful, if you could help me to accomplish my task.

Kind Regards,

Marie Blizzard

Constraints

- Your **Deque** must be implemented with **LinkedList<T>**.
- Your **Deque** class must inherit the given **IDeque<T>** interface and implement all of its methods and properties.
- Your **Deque** implementation must place no fixed limits on the number of elements it may contain.
- Your source code should be very well documented.
- Your source code should be written in the principles of high quality code.

*Tip (not provided from the client)

In computer science, **double-ended queue** ("dequeue", often abbreviated to "**deque**", pronounced "**deck**") is an abstract data type - a linear collection that supports element insertion and removal at both ends. Basically, the **Deque** combines the functionality of both queue and stack data structures. Said formally, **Deque** generalizes a queue, for which elements can be added to or removed from either the **front** (head) or **back** (tail). Normal **Deque** implementations place no fixed limits on the number of elements they may contain.

You are given an interface **IDeque<T>** consisting of the following methods and properties:

- **int Count** – returns the number of elements in the **Deque**;
- **void PushFirst(T element)** - inserts element at the beginning of the **Deque**;
- **void PushLast(T element)** - inserts element at the end of the **Deque**;
- **T PopFirst()** - removes and returns the element at the beginning of the **Deque**;
- **T PopLast()** - removes and returns the element at the end of the **Deque**;
- **T PeekFirst()** - returns the element at the beginning of the **Deque** without removing it;

- **T PeekLast()** - returns the element at the end of the **Deque** without removing it;
- **void Clear()** - clears the **Deque**. It will be empty after this operation completes;
- **bool Contains(T element)** - returns true if this collection contains the specified element;

Task 2 – Help Resource

Let's assume there are 5 more clients with similar request. Your task is to prepare a reusable help resource (Code Library) where the approach is explained and a project is attached, so you can directly navigate clients to it.