namespace \_09.Collection\_Hierarchy.Models

{

using Interfaces;

using System.Collections.Generic;

public class AddCollection<T> : IAddCollection<T>

{

public AddCollection()

{

this.Data = new List<T>();

}

protected List<T> Data { get; set; }

public virtual int Add(T element)

{

this.Data.Add(element);

return this.Data.Count - 1;

}

}

}

==================================================================

namespace \_09.Collection\_Hierarchy.Models

{

using Interfaces;

using System.Linq;

public class AddRemoveCollection<T> : AddCollection<T>, IAddRemoveCollection<T>

{

private const int IndexForAddingNewElement = 0;

public virtual T Remove()

{

var lastElement = this.Data.Last();

this.Data.RemoveAt(this.Data.Count - 1);

return lastElement;

}

public override int Add(T element)

{

this.Data.Insert(IndexForAddingNewElement, element);

return IndexForAddingNewElement;

}

}

}

===============================================================

namespace \_09.Collection\_Hierarchy.Models

{

using System.Collections.Generic;

using Interfaces;

using System.Linq;

public class MyList<T> : AddRemoveCollection<T>, IMyList<T>

{

private const int RemovalIndex = 0;

public IReadOnlyCollection<T> Used

{

get

{

return this.Data as IReadOnlyCollection<T>;

}

}

public override T Remove()

{

var firstElemennt = this.Data.First();

this.Data.RemoveAt(RemovalIndex);

return firstElemennt;

}

}

}