Lists

Simon Robinson
http://TechieSimon.com
@TechieSimon


Lists
Linked Lists,
Stacks and Queues
Dictionaries
Sets
I will cover only the most
important collection types

Module Overview

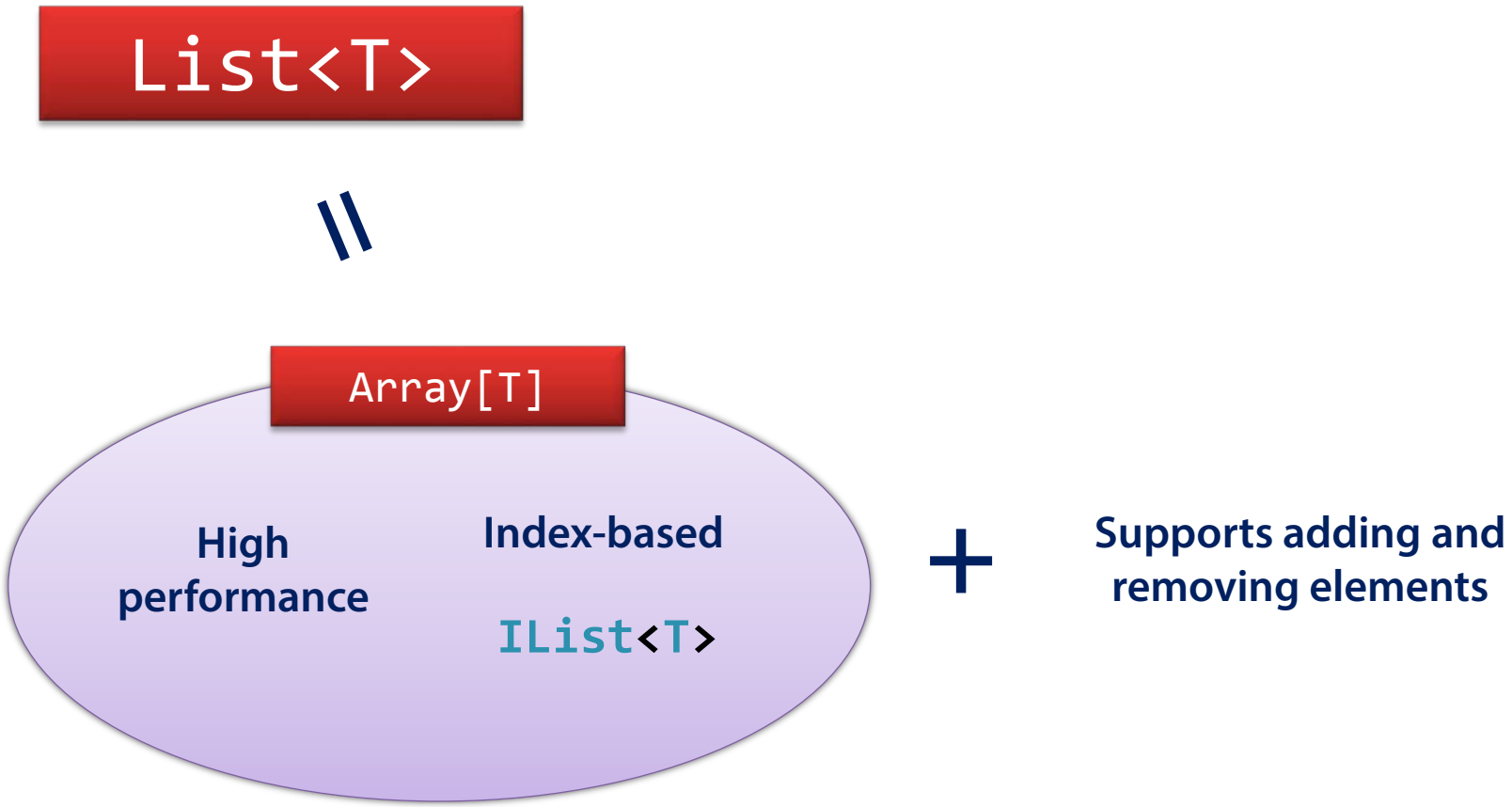
- `List<T>`
  - Extensive API
  - Like array but with adding/removing elements
- `ReadOnlyCollection<T>`
  - Read-only wrapper for lists
- `Collection<T>`
  - Allows lists to be customized
- `ObservableCollection<T>`
  - List with change notifications

Module Overview

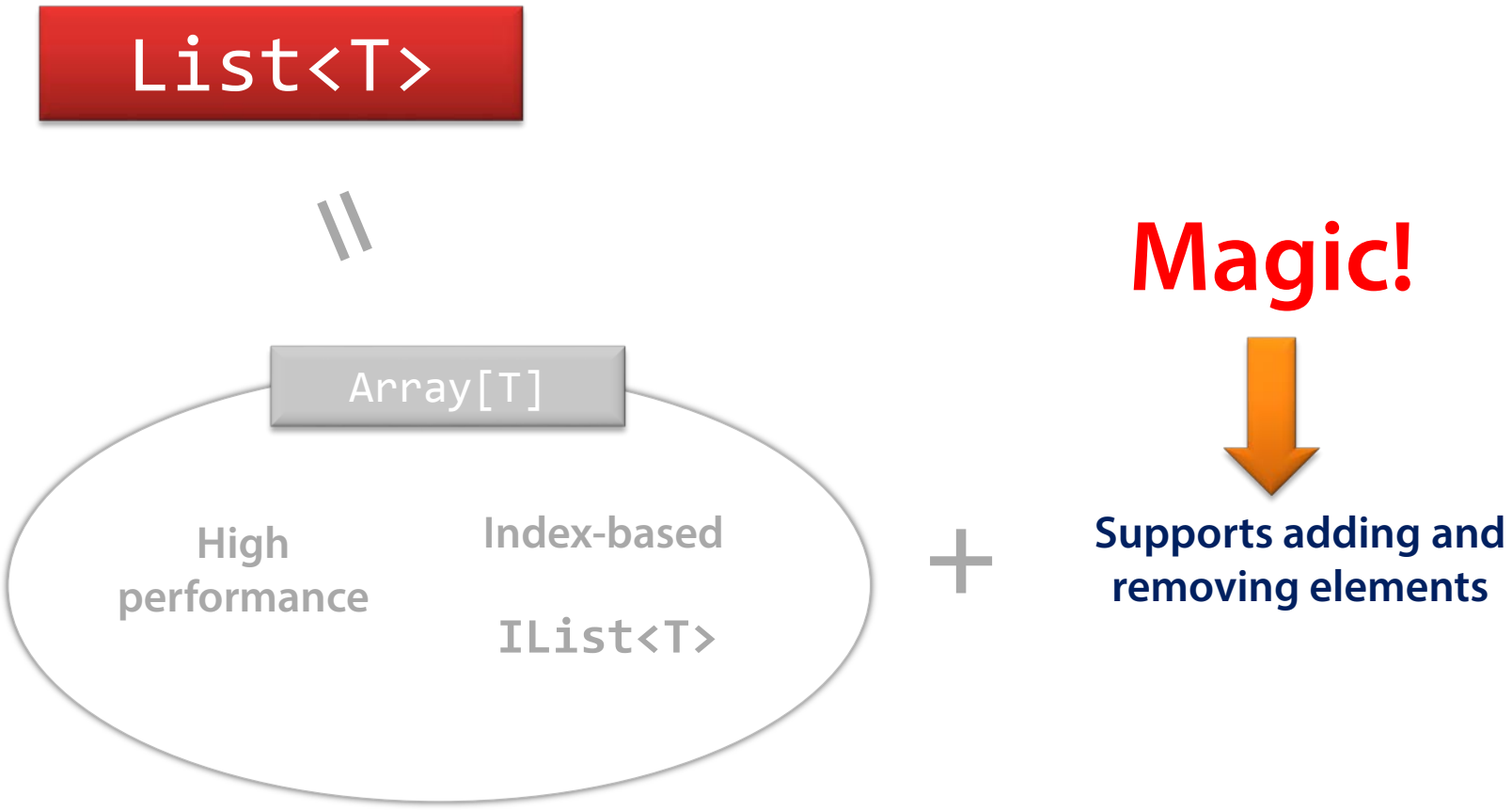
- `List<T>`
  - Extensive API
  - Like array but with adding/removing elements
- `ReadOnlyCollection<T>`
  - Read-only wrapper for lists
- `Collection<T>`
  - Allows lists to be customized
- `ObservableCollection<T>`
  - List with change notifications

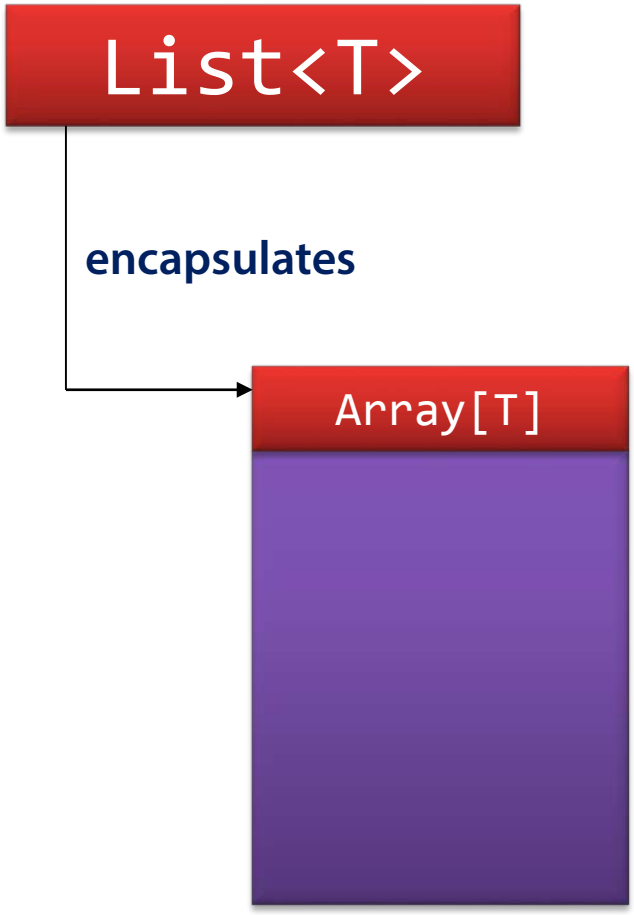
```
using System.Collections.Generic;
```

```
using System.Collections.ObjectModel;
```

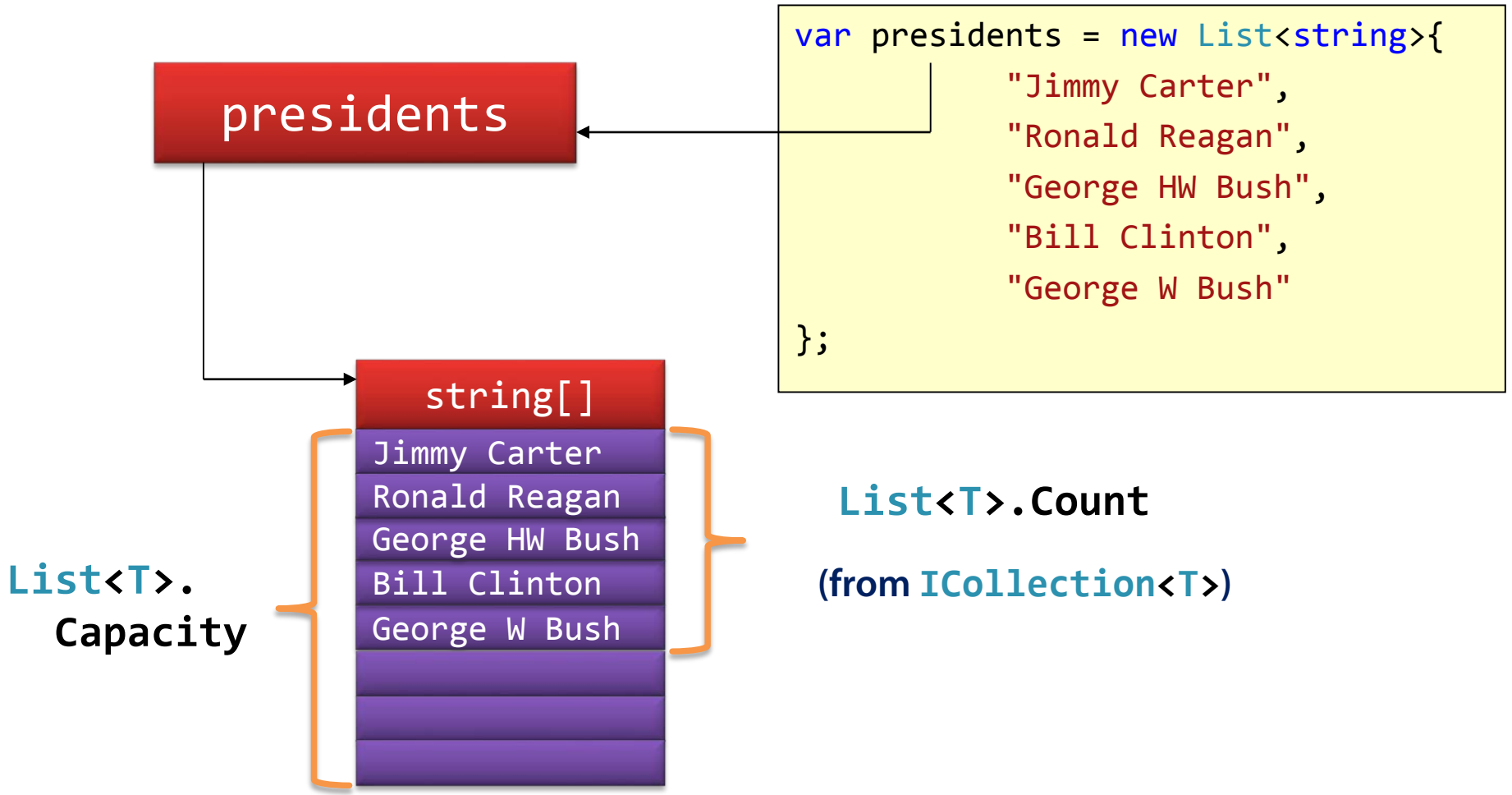

List<T>
=
Array[T]
High performance
Index-based
IList<T>
+
Supports adding and removing elements

Code Demo

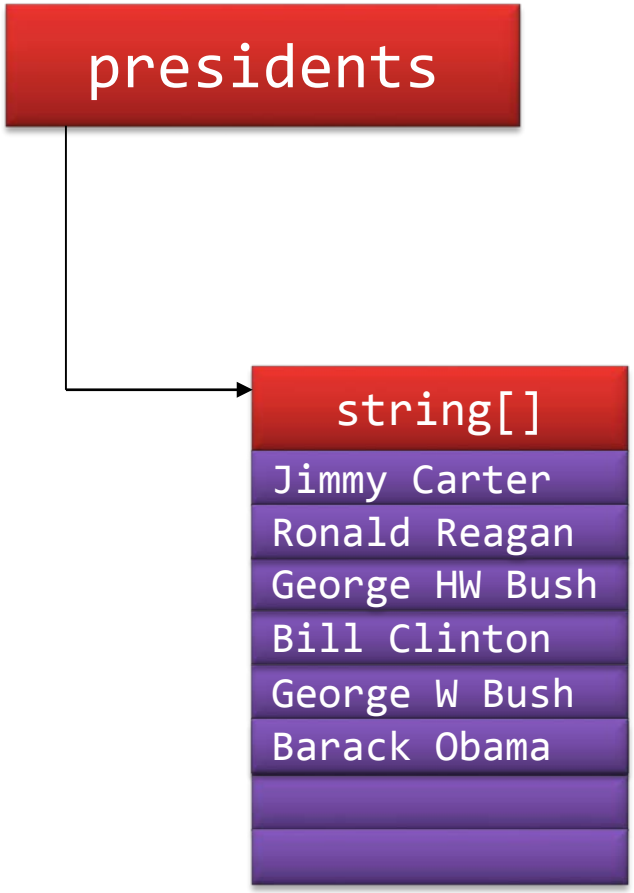

List<T>
\\
Array[T]
High performance
Index-based
IList<T>
+
Magic!
Supports adding and removing elements




```
var presidents = new List<string>{
            "Jimmy Carter",
            "Ronald Reagan",
            "George HW Bush",
            "Bill Clinton",
            "George W Bush"
};
```


presidents
var presidents = new List<string>{
"Jimmy Carter",
"Ronald Reagan",
"George HW Bush",
"Bill Clinton",
"George W Bush"
};
string[]
Jimmy Carter
Ronald Reagan
George HW Bush
Bill Clinton
George W Bush
List<T>.
Capacity
List<T>.Count
(from ICollection<T>)
Bigger than required to
hold the elements

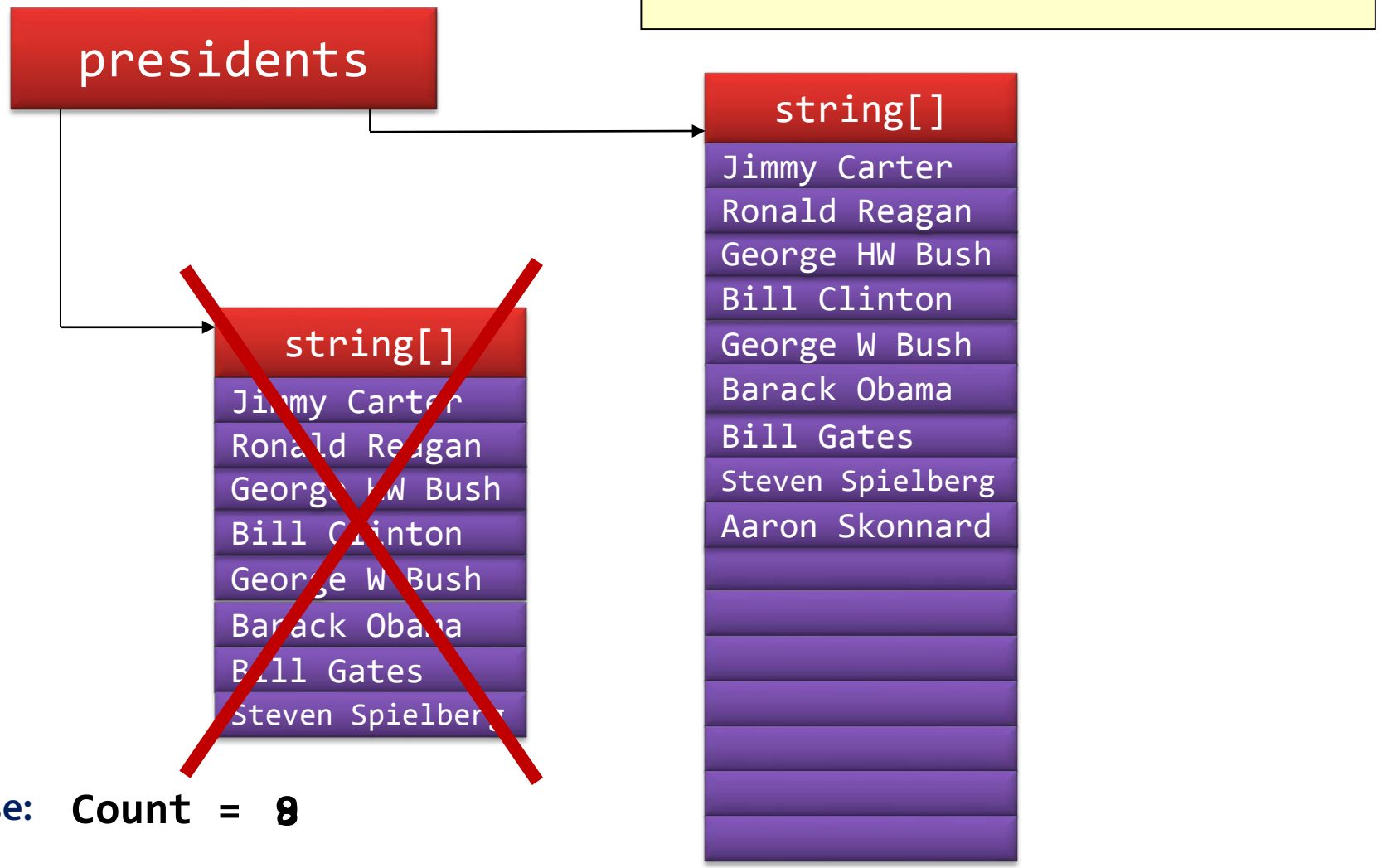
```
presidents.Add("Barack Obama");
```



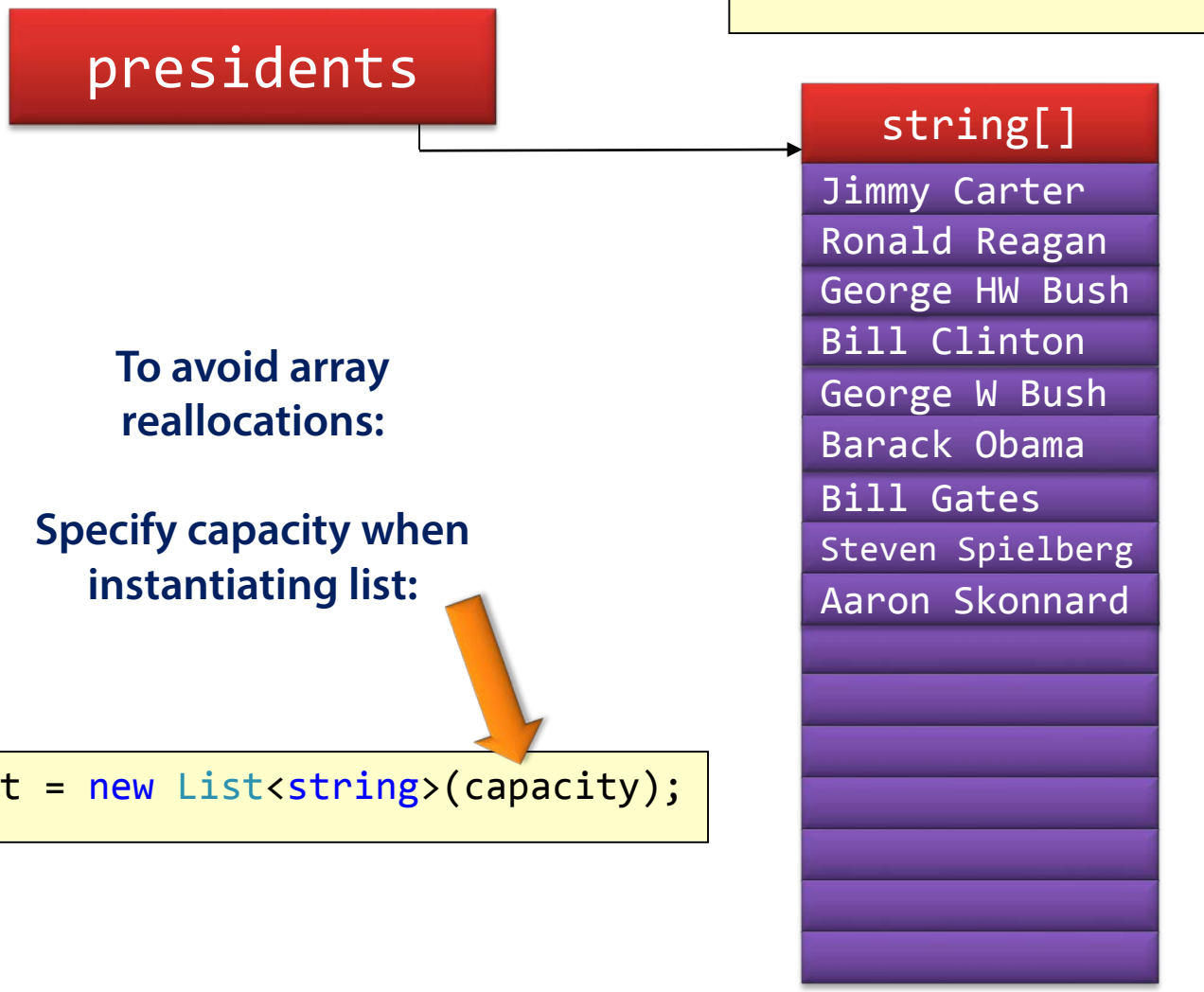
In use: **Count = 6**

Code Demo

```
presidents.Add("Aaron Skonnard");
```



```
presidents.Add("Aaron Skonnard");
```



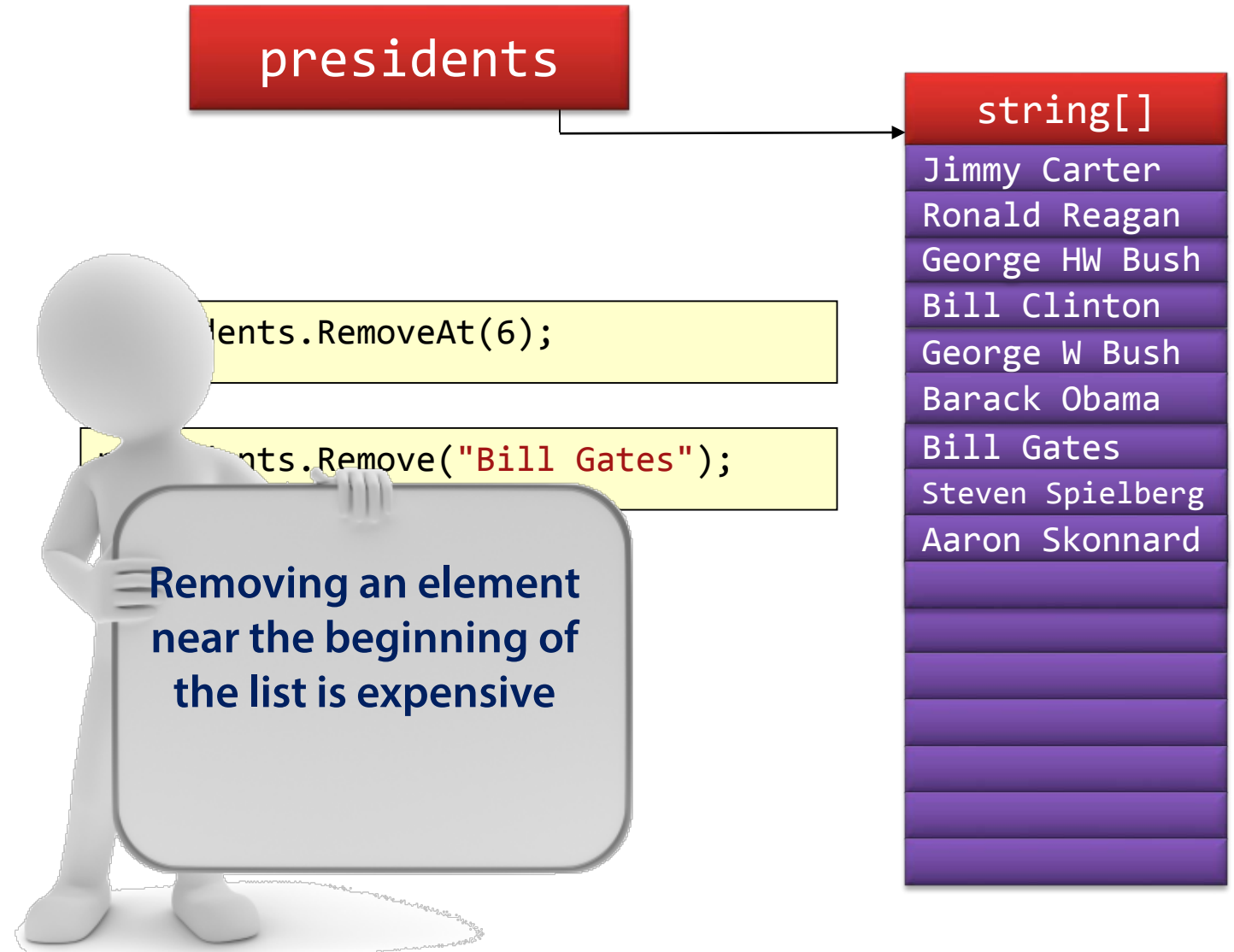
To avoid array reallocations:

Specify capacity when instantiating list:

```
var lst = new List<string>(capacity);
```

Code Demo

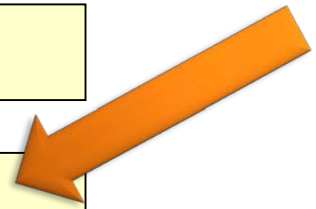
Removing Elements



Removing Elements

```
presidents.RemoveAt(6);
```

```
presidents.Remove("Bill Gates");
```



This will take longer than `RemoveAt(6)`

- Must find index by searching elements

- Then `RemoveAt(index)`

List<T> Capabilities

IList<T>

- Access elements by index

What can it do?

List<T>

ReadOnlyCollection<T>

```
List<T>.AsReadOnly()
```

Gives read-only access to a list

Code Demo

Beware!

Underlying list can still be modified by:

```
using System.Reflection;
```

A direct reference:

Individual elements of the list may be modifiable!

Collection<T>

Provides an implementation of

List<T>: ✔ Lightweight

✔ Efficient

✘ Not customizable

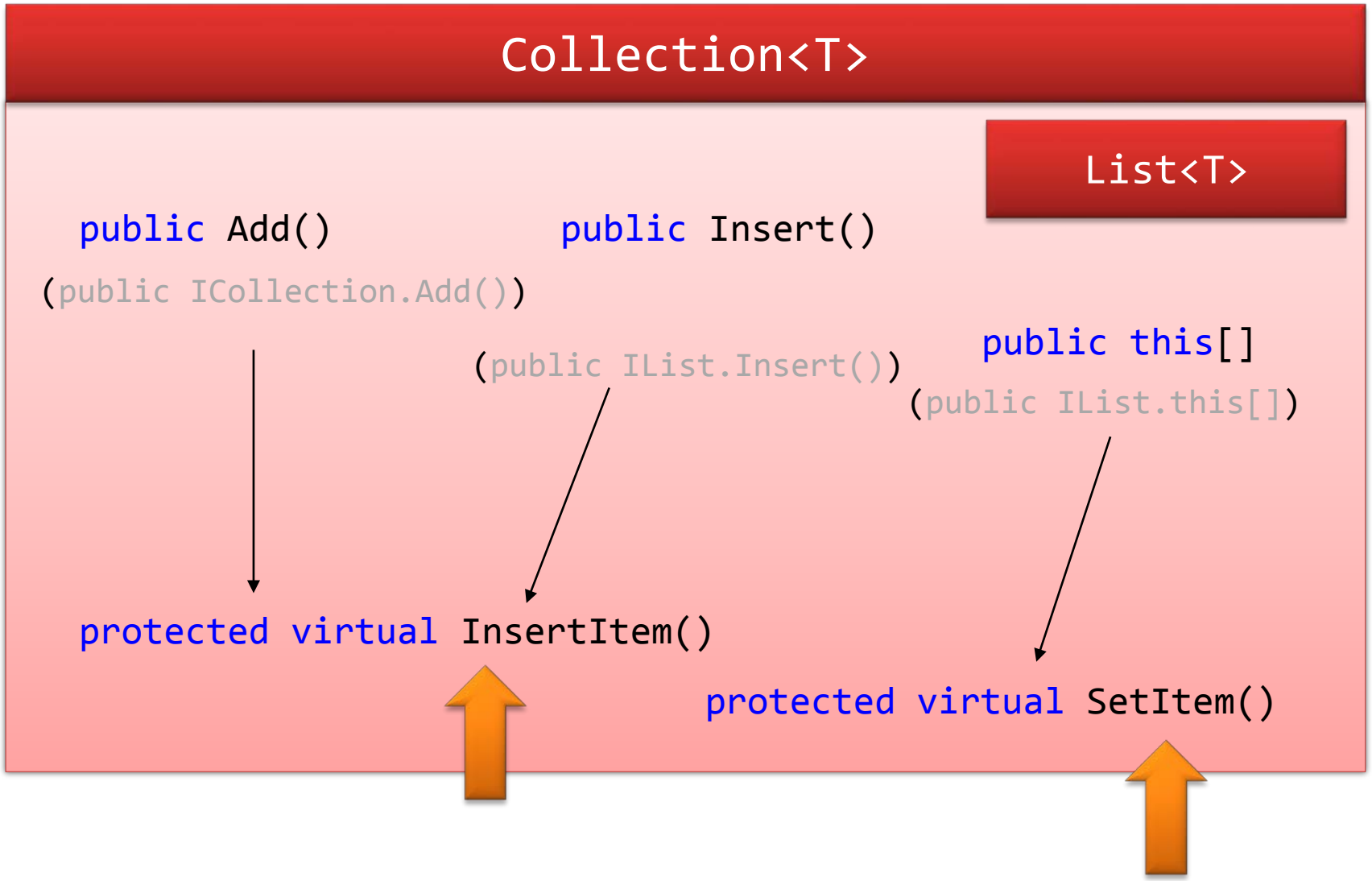
Collection<T>: ✔ Customizable

Doesn't List<T> do that already?

Example:

This list must only accept non-whitespace, non-null strings

Code Demo

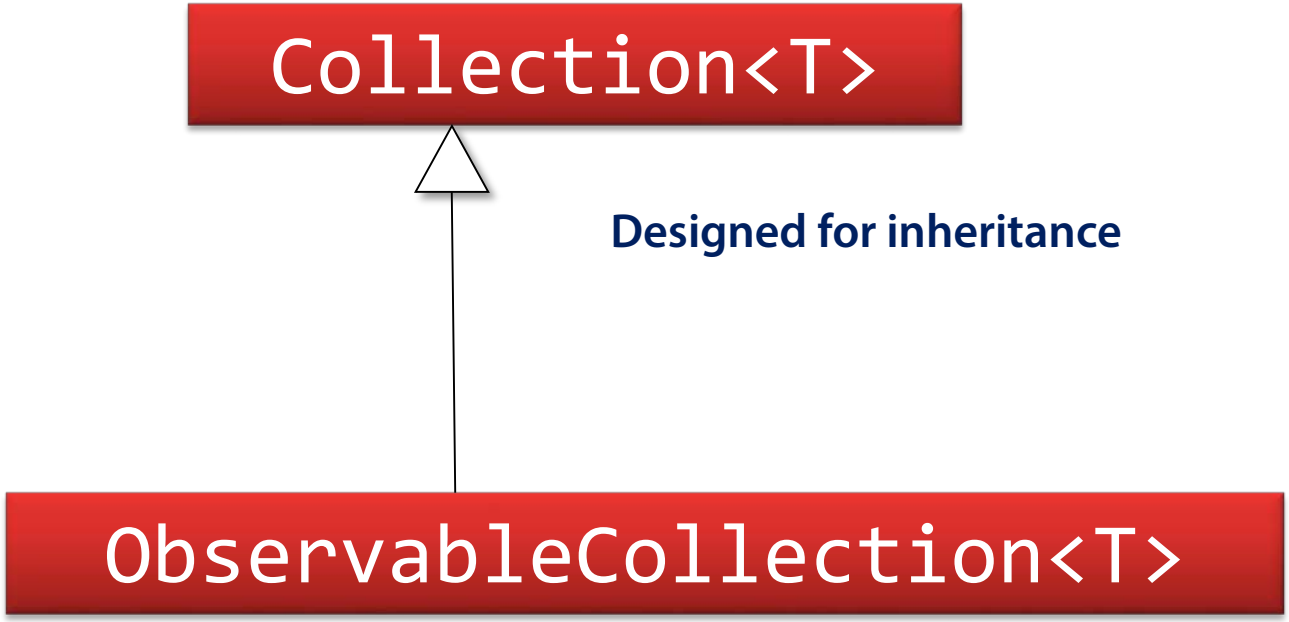

Collection<T>
List<T>
public Add()
(public ICollection.Add())
public Insert()
(public IList.Insert())
public this[]
(public IList.this[])
protected virtual InsertItem()
protected virtual SetItem()

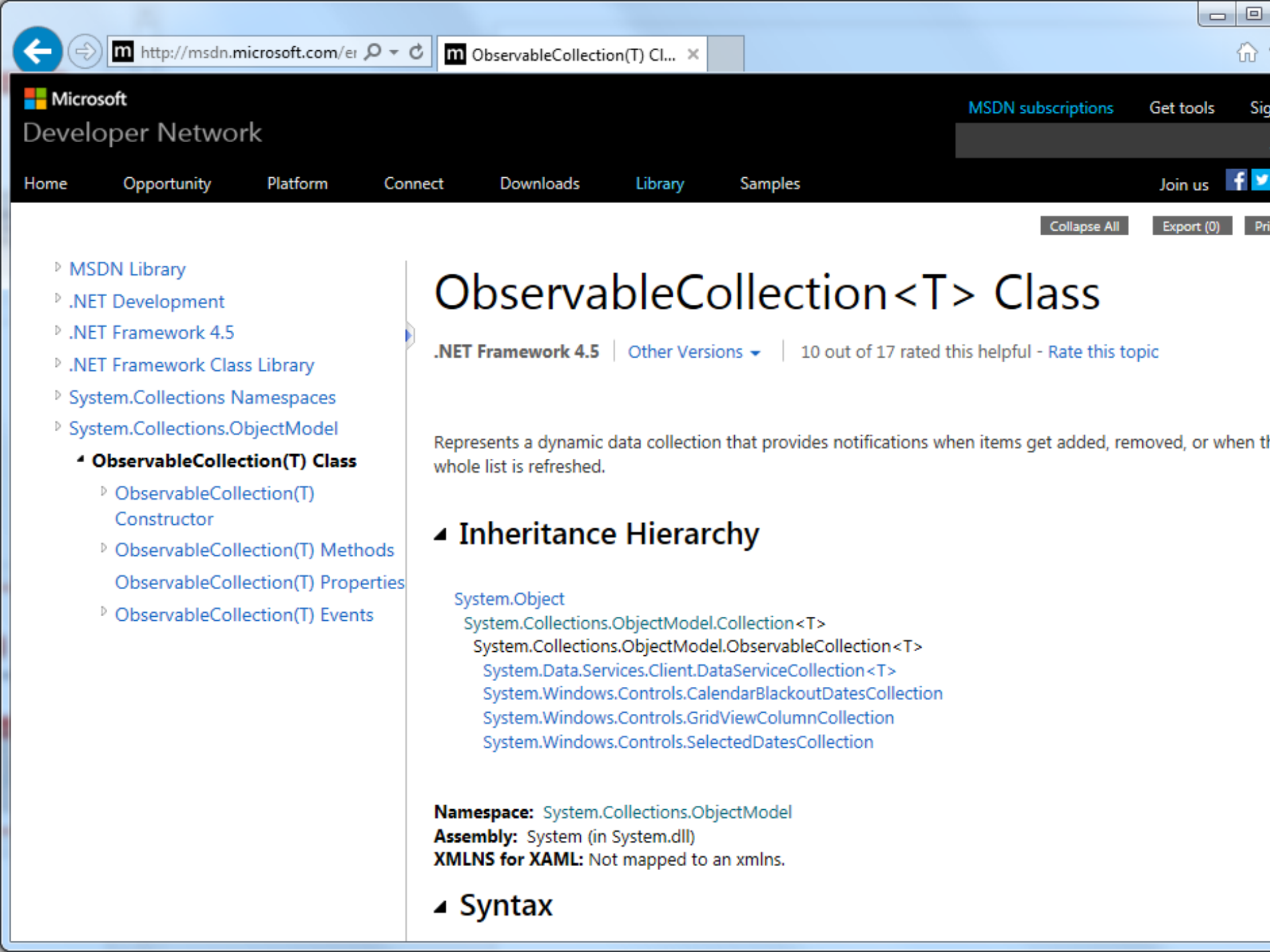
Code Demo

ObservableCollection<T>

List that provides change notifications

Code Demo


Collection<T>
Designed for inheritance
ObservableCollection<T>


http://msdn.microsoft.com/er
ObservableCollection(T) Cl...
Microsoft
Developer Network
MSDN subscriptions
Get tools
Sig
Home
Opportunity
Platform
Connect
Downloads
Library
Samples
Join us
Collapse All
Export (0)
MSDN Library
.NET Development
.NET Framework 4.5
.NET Framework Class Library
System.Collections Namespaces
System.Collections.ObjectModel
ObservableCollection(T) Class
ObservableCollection(T) Constructor
ObservableCollection(T) Methods
ObservableCollection(T) Properties
ObservableCollection(T) Events
ObservableCollection<T> Class
.NET Framework 4.5
Other Versions
10 out of 17 rated this helpful - Rate this topic
Represents a dynamic data collection that provides notifications when items get added, removed, or when th
whole list is refreshed.
Inheritance Hierarchy
System.Object
System.Collections.ObjectModel.Collection<T>
System.Collections.ObjectModel.ObservableCollection<T>
System.Data.Services.Client.DataServiceCollection<T>
System.Windows.Controls.CalendarBlackoutDatesCollection
System.Windows.Controls.GridViewColumnCollection
System.Windows.Controls.SelectedDatesCollection
Namespace: System.Collections.ObjectModel
Assembly: System (in System.dll)
XMLNS for XAML: Not mapped to an xmlns.
Syntax

Summary – Lists

- `List<T>`
  - Encapsulates `T[]`
  - Rich functionality
  - Allows adding and removing elements
- `Collection<T>`
  - Allows customizing the behaviour of lists
  - Eg. Change notifications: `ObservableCollection<T>`
- `ReadOnlyCollection<T>`
  - Gives read-only access to a collection
  - Customizable