Adding and Removing Data with List<T>

Simon Robinson
Software Developer

@TechieSimon www.SimonRobinson.com

Overview

Arrays don't support adding/removing

- Fixed size

Leads to `List<T>`

- Does support adding/removing

Demo

Displays bus routes in an array

- Unwanted data must be removed first

Arrays are fixed size collections

Instantiating an Array

List the items

```
BusRoute[] routes = {
    new BusRoute(40, // etc.
    new BusRoute(42, // etc.
    // etc.
};
```

Specify how many items

```
BusRoute[] routes
        = new BusRoute[4];
```

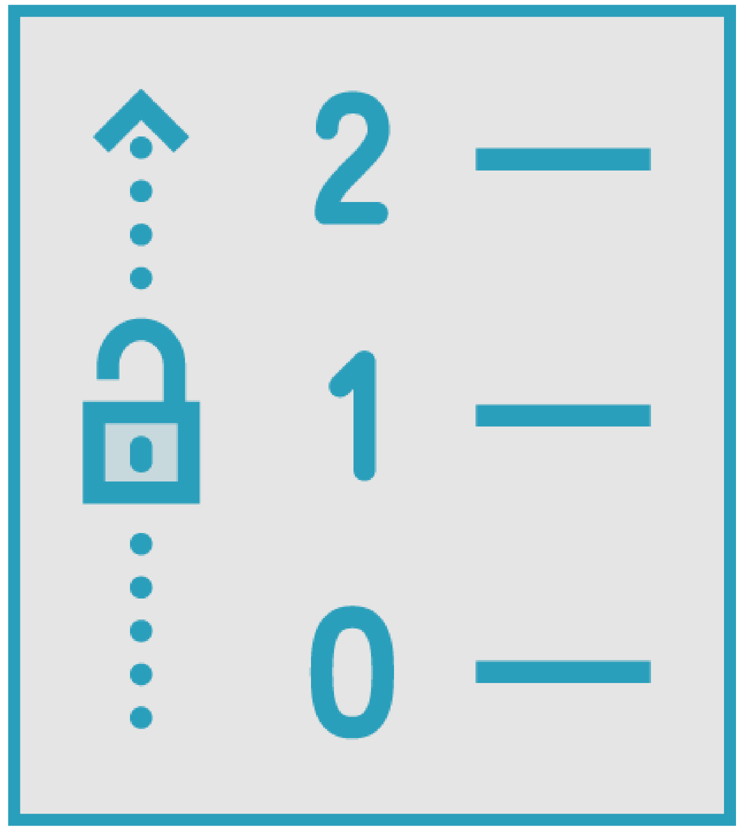
⚠ You can’t subsequently change the array size!

Comparing Arrays and Lists

`List<T>`

Not a fixed size

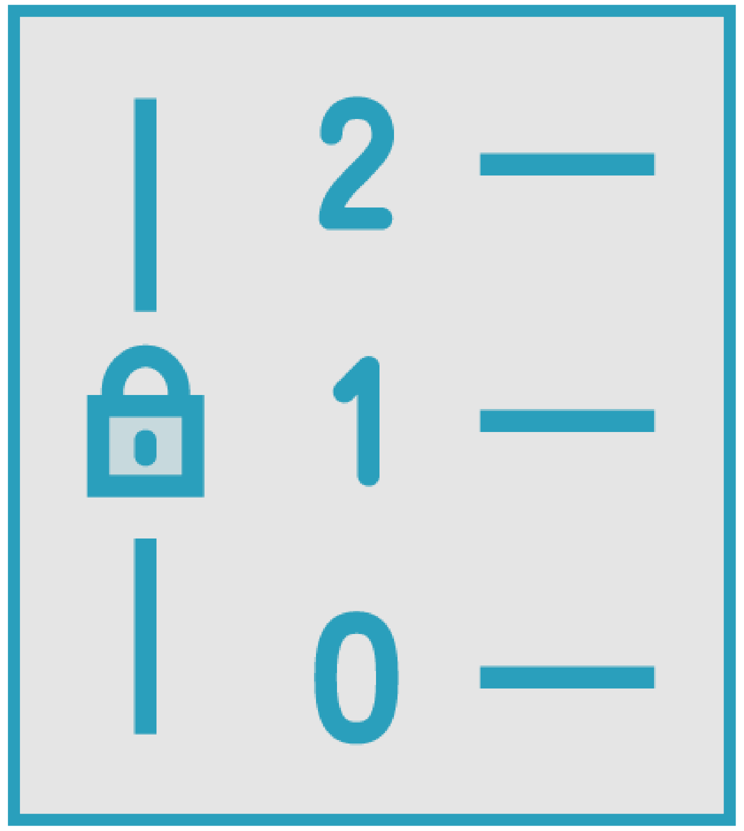
Instantiate empty, then add the elements you need



`T[]`

Fixed size

Instantiate with the fixed number of elements



Initializers

Array initializer

```
BusRoute[] routes = {
    // list the items here
};
```

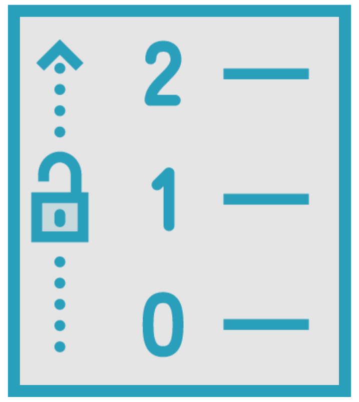
Creates an array with the items

Collection (list) initializer

```
List<BusRoute> routes = new List<BusRoute> {
    // list the items here
};
```

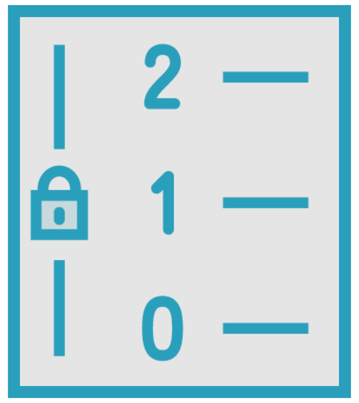
Expands to code that creates an empty list
Then adds the items

Lists and Arrays



`List<T>`

- Extra ability to add and remove items



`T[]`

- Built into the language

Use a list if you need to add and remove items, an array if you don’t

Caution!

`List<T>`: Adding and removing can be slow!

`List<T>` stores data in an internal array

It has to move data around the array as you add/remove items

It might even copy the data to a new internal array

Use mainly for look-up and enumeration

Arrays and lists are super-fast for lookup/enumeration

Summary

Arrays are a fixed size

- Can't add or remove elements
- `Array.Resize()` actually copies the array

Lists

- Similar to arrays but not fixed size
- But not optimized for frequent adding/removing

Up Next: Dictionaries