

CSSE2002/7023

Semester 1, 2021

Programming in the Large

Week 3.3: Intro to Collections

In this Session

- Collections
- `java.util.Stack`
- Primitive Wrappers

Why Not Arrays?

Arrays aren't great for everything:

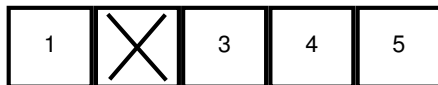
- Fixed size at creation time
 - Do you know how much space you are going to need?
- Don't automatically close gaps



Why Not Arrays?

Arrays aren't great for everything:

- Fixed size at creation time
 - Do you know how much space you are going to need?
- Don't automatically close gaps



Why Not Arrays?

Arrays aren't great for everything:

- Fixed size at creation time
 - Do you know how much space you are going to need?
- Don't automatically close gaps

1	3	3	4	5
---	---	---	---	---

Why Not Arrays?

Arrays aren't great for everything:

- Fixed size at creation time
 - Do you know how much space you are going to need?
- Don't automatically close gaps

1	3	4	4	5
---	---	---	---	---

Why Not Arrays?

Arrays aren't great for everything:

- Fixed size at creation time
 - Do you know how much space you are going to need?
- Don't automatically close gaps

1	3	4	5	5
---	---	---	---	---

Why Not Arrays?

Arrays aren't great for everything:

- Fixed size at creation time
 - Do you know how much space you are going to need?
- Don't automatically close gaps

1	3	4	5	?
---	---	---	---	---

Other Options

We will talk about some other data structures. We are looking at their API (what they can do) not (unless otherwise indicated) their performance (how fast they do it)¹.

- Stack
- List
- Set
- Map

All of these live in `java.util.*`.

¹See an algorithms course (i.e. COMP3506) for that.

Stack (L.I.F.O/F.I.L.O)

<code>empty()</code>	Is this stack empty?
<code>peek()</code>	Return the object at the top of the stack.
<code>pop()</code>	Remove the object at the top of the stack and return it.
<code>push(obj)</code>	Put obj on the top of the stack.

Stack — Generics

Originally Java collections stored Objects.

We often want to be more restrictive.

- So we don't get Cars in our stack of Cats.

To declare a stack of Strings:

```
Stack<String> s = new Stack<String>();
```

OR

```
Stack<String> s = new Stack<>();
```

The compiler can determine that it needs to create a stack that only accepts String objects.

Stack

StackDemo.java

What is it good for?

- Reversing things?
- Putting something aside to come back to later while you deal with something else now.
- “Depth-first” algorithms

Primitives in Collections

What about `Stack<int>`?

²In `java.lang`

Primitives in Collections

What about `Stack<int>`?

- Collections only store objects

Java has a class² for each primitive type:

`Boolean`, `Byte`, `Character`, `Double`,
`Float`, `Integer`, `Long`, `Short`

So you can have `Stack<Integer>`

See: `IntStack.java`

Autoboxing — create objects in the background

- Primitives are not objects

²In `java.lang`