Generic Interfaces and Classes

Brice Wilson

@brice_wilson www.BriceWilson.net

Creating a Generic Interface and Class

Stack Data Structure

Last in, First out

Stack.ts

Stack Data Structure

Last in, First out

Stack.ts

```
numberStack.push(42);
```

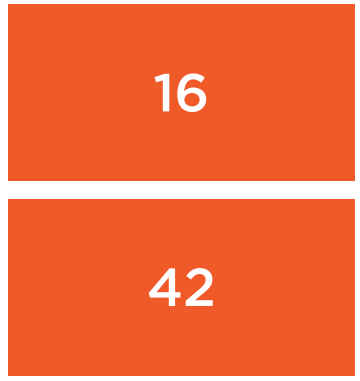
Stack Data Structure

Last in, First out

Stack.ts

```
numberStack.push(42);

numberStack.push(16);
```



Stack Data Structure

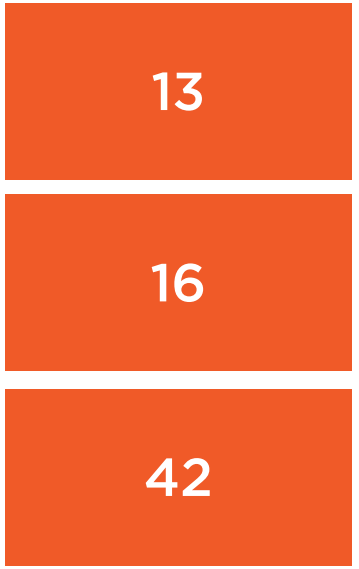
Last in, First out

Stack.ts

```
numberStack.push(42);

numberStack.push(16);

numberStack.push(13);
```

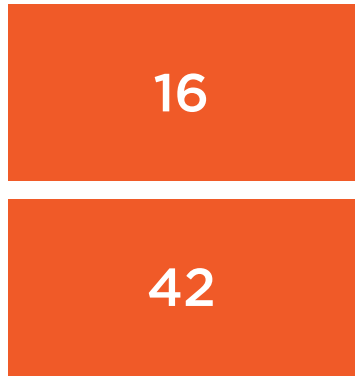


Stack Data Structure

Last in, First out

Stack.ts

```
numberStack.push(42);
numberStack.push(16);
numberStack.push(13);
numberStack.pop();
```

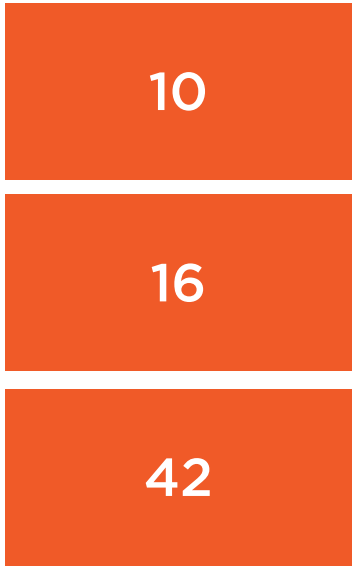


Stack Data Structure

Last in, First out

Stack.ts

```
numberStack.push(42);
numberStack.push(16);
numberStack.push(13);
numberStack.pop();
numberStack.push(10);
```

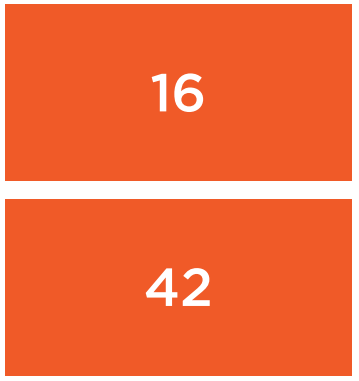


Stack Data Structure

Last in, First out

Stack.ts

```
numberStack.push(42);
numberStack.push(16);
numberStack.push(13);
numberStack.pop();
numberStack.push(10);
numberStack.pop();
```



Stack Data Structure

Last in, First out

Stack.ts

```
numberStack.push(42);
numberStack.push(16);
numberStack.push(13);
numberStack.pop();
numberStack.push(10);
numberStack.pop();
numberStack.pop();
```

Demo

Implementing a generic interface with a generic class

Demo

Applying type constraints to a generic class

Generic classes are only
generic over their instances.

Summary

- Use generic interfaces with object literals or generic classes
- Generic classes offer type-safe versatility (with or without implementing an interface)
- Type constraints increase practicality