

make this into an `OrderedArrayList`  
 add all of these numbers to the  
`OrderedArrayList` using functions  
 from `OrderedArrayList.java`  
 (use the same algorithm as `inOrder`)

5 6 7 8 10 4 3 15 22 1  
 5 6 7 8 4 10 3 15 22 1

orderedBefore = 5

5 6 7 8 4 3 10 15 22 1

find the index for which  $\_arr[i] < 5$   
 $< \_arr[i+1]$

when finished, increment  
 orderedBefore

in `ArrayList()`

in `OrderedArrayList()`

`add(int value)`

`addO( int value)`

0. Find  $i$  for which
1. Insert the value at that index  
 using `add( int index, int value)`

constructor

0. Make an `ArrayList` which will be an instance  
 variable for each instance of `OrderedArrayList`

`add(int index, int value)`

(no such function should exist in `OrderedArrayList`)

`toString`

(additional constructor taking an  
`ArrayList`, which may or may not be  
 in order, as an argument, and  
 putting it in order using the `add()`  
 defined in `OrderedArrayList()`)

Take an `ArrayList`

`remove(int index)` (Can work identically)

`get(int index)` (Can work identically)

`set(int index, int value)`  
`replaceO( int value, int  
 newValue)`

`find( int value)`

Q. Would it be more useful to  
 change these methods to only  
 use values, instead of indexes?

You are screen sharing Stop Share

[3 4 5] 2

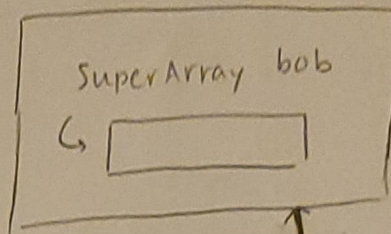


Foo  
- SuperArray - arr

+ order (SA a)

SuperArray  
- size  
- data  
+ int add (int val)  
+ int addAtIndex (int index, int newVal)  
+ void remove  
+ void set (int index, int newVal)  
+ void add (int newVal)  
+ void addAtIndex (int index, int newVal)  
+ void remove

Foo berty



uses

Clear

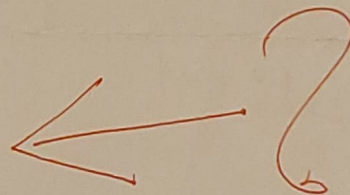
ListInt  
interface  
+ add (int val)  
+ addAtIndex  
+ remove  
+ set

Q. Do we need toString for Foo?

Q. How does interface work with wrapper class Foo & how do we represent it in a diagram?

Q. Should this be a ListInt or a SuperArray?

What's the algorithm for order?



Alg. for ordering;

① Check if  $x_i < x_{i+1}$

if yes

↳ do nothing

1 2 2 3 5 8

2 3 5 8 1 2

maintaining an invariant -  
it never is in a different state.

Algo for order:

- ① We check if  $AL[i] \leq AL[i+1]$   
if no  
↳

5 8 7 3 2 9  
↳ 5 ~~8~~ 7 8 - - -

5 7 3 8 - - -

5 7 3 2 8 - - -

5 7 3 2 8 9 - -



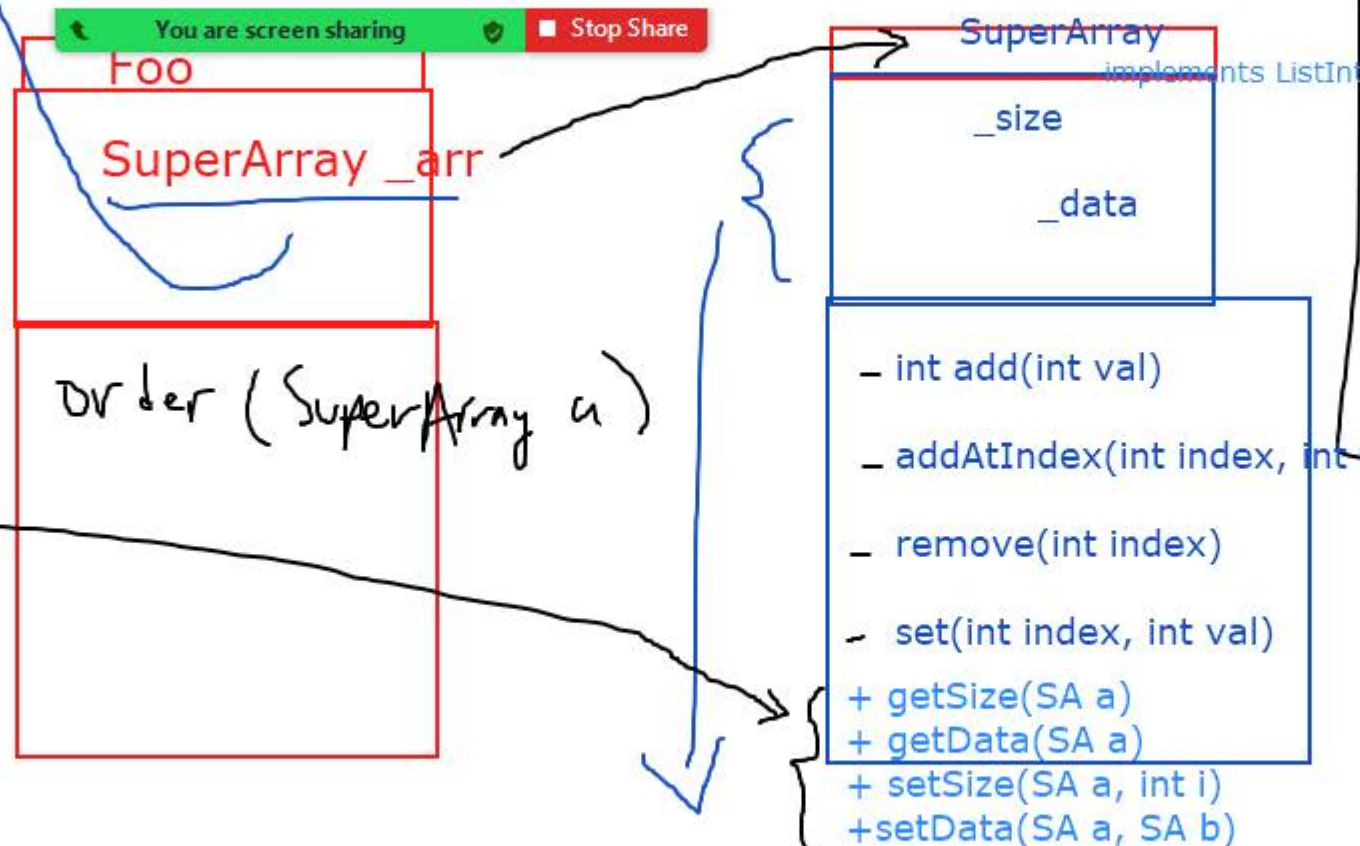
### 3 Lucky Duckies: Gloria Lee, Ziying Jian, Nora Miller

Should this be a ListInt or a SuperArray?

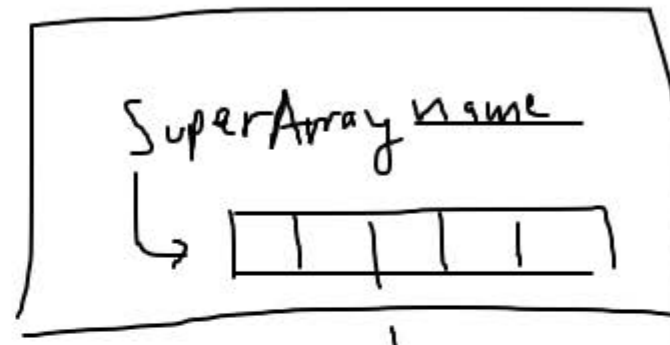
- Maybe var type is

inst  
var

may or may not be necessary, depending on whether works



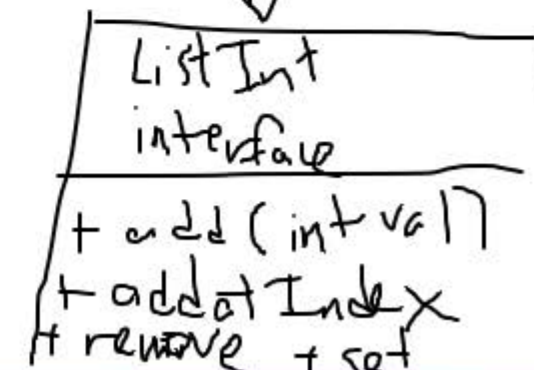
Foo name



uses

Q. do we need toString() for Foo?

A. In



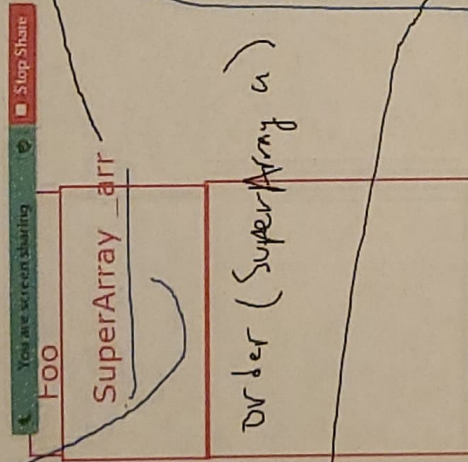
### 3 Lucky Duckies: Gloria Lee, Ziying Jian, Nora Miller

Should this be a ListInt or a SuperArray?

- Maybe var type is

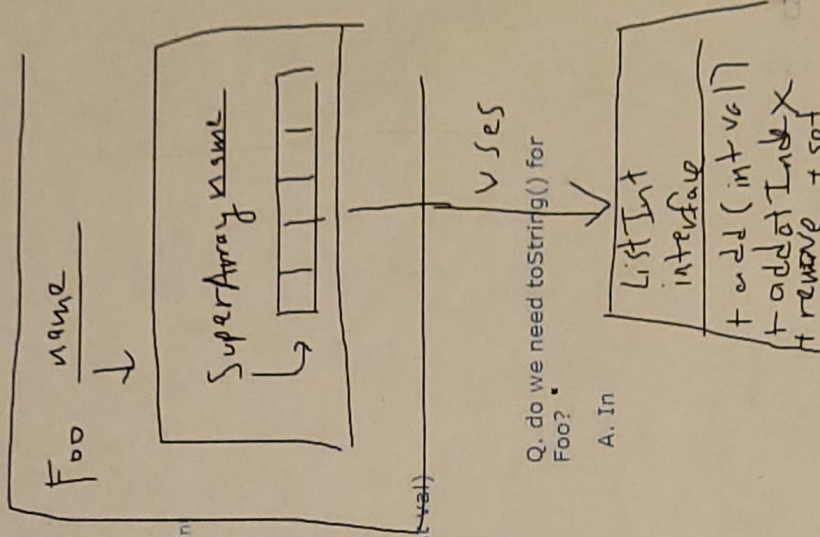
inst  
var

may or may not be necessary, depending on whether works



Order (SuperArray a)

can be accessed by \_arr.\_size, for example



Q. do we need toString() for Foo?

A. In

ListInt interface  
+ add (int val)  
+ addAtIndex  
+ remove + set