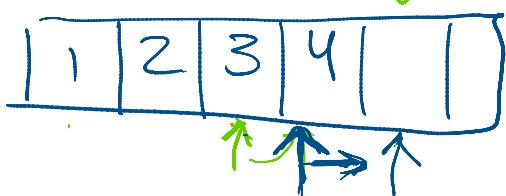


Cursors

- move through a sequence
- current element
- data structure:
 - current index
 - isCurrent
- part of the sequence ADT
 - any time you change the cursor, you change the sequence

→ behaviors:

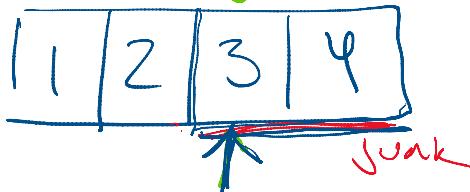
- advance, start make next non-current current
- remove current
- getCurrent



currentIndex = 2

isCurrent = true

atEnd
→ verifies for user at end
size



isCurrent = false

advance

many items = 2

currentIndex = 2

invariant: many items == cI
→ no currentIterators

- Data Structure:
 - Current Index
 - isCurrent
 - my Version *

→ Behaviors:

- * iterator is an interface
- * abstract methods
 - not implemented
 - must implement

→ Behaviors:

→ hasNext:

→ lets you know if there
are more elements
*must be implemented
→ no default

→ next:

→ goes to the next
current
*must be implemented
→ returns current

→ remove:

→ removes current
*does not have to be
implemented

*Fail Fast: can go stale

→ change collection
w/out using the iterator

Collection: 

→ add(e): all go stale

*iterators throw
Concurrent Mod. Exception

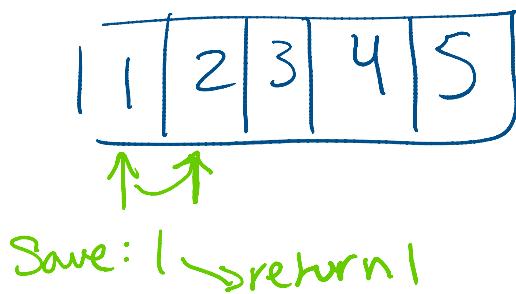
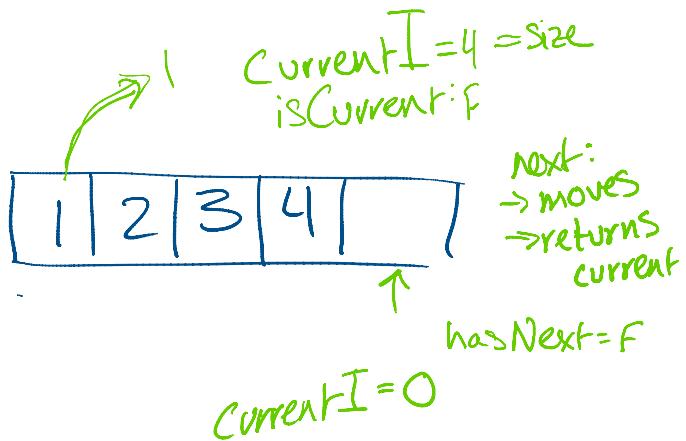
*

→ iterator is a nested
class

*non-static

- can't see outside of

→ not implemented
→ you must implement



next:
→ saves current
→ move
→ return saved element

→ call collection's remove:
→ all iterators go stale
*clear *collection's version

→ i.remove()
→ i2 goes stale
*change both versions
*be the same



iterator class

.. Iterator

* non-static
 * → can see outside of
 * itself
 * private
 * user never
 $it = new MyIterator()$
 → collection: iterator()

iterator class
 private class MyIterator implements Iterator<?>
 public MyIterator()
 → sets fields
 → checks well-formed

enhanced for loop

for(<type> <variable>: <list/collection>)

~~for(String element: data)~~

* Iterator $it = new Iterator()$

for(; it.hasNext();)
 element = it.next()

while (it.hasNext())
 element = it.next()