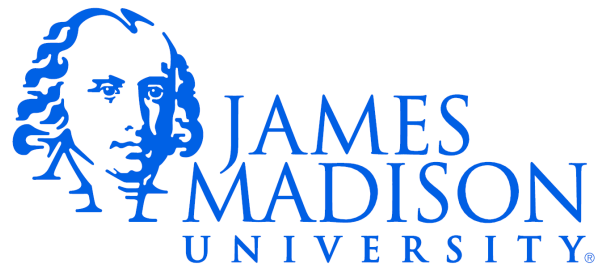# CS159

Reminder

- Naming convention for Java Collection types: ArrayList

  – Array – Coded using Arrays "under the hood"

  – List – Implements the List Interface

  An ordered collection (also known as a sequence). The user of this interface has precise control over where in the list each element is inserted. The user can access elements by their integer index (position in the list), and search for elements in the list
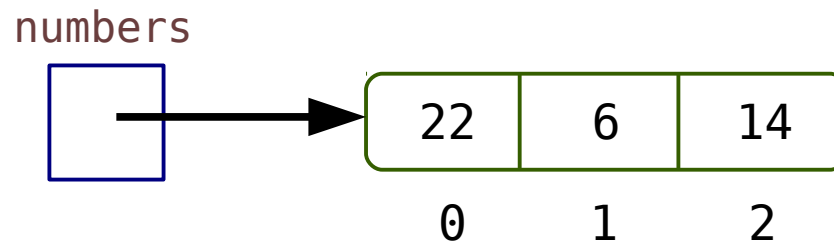
Question for Today

- Arrays are great. Why would we want any other implementation?

- In general: why would Java provide multiple implementation for their collection interfaces? Why not just pick the best one?

Why Arrays Are Great (And Not Great)

- Arrays store elements in one contiguous block of memory:

```java
int[] numbers = {22, 6, 14};
```

numbers

22	6	14
0	1	2

- Advantages of arrays:
  - Very fast access to elements by index.
- Disadvantages of arrays:
  - Let's look at an example...
  - ArraysAreBad.java

Exercise...

- Trace the execution of the following code snippet. Draw the contents of memory.

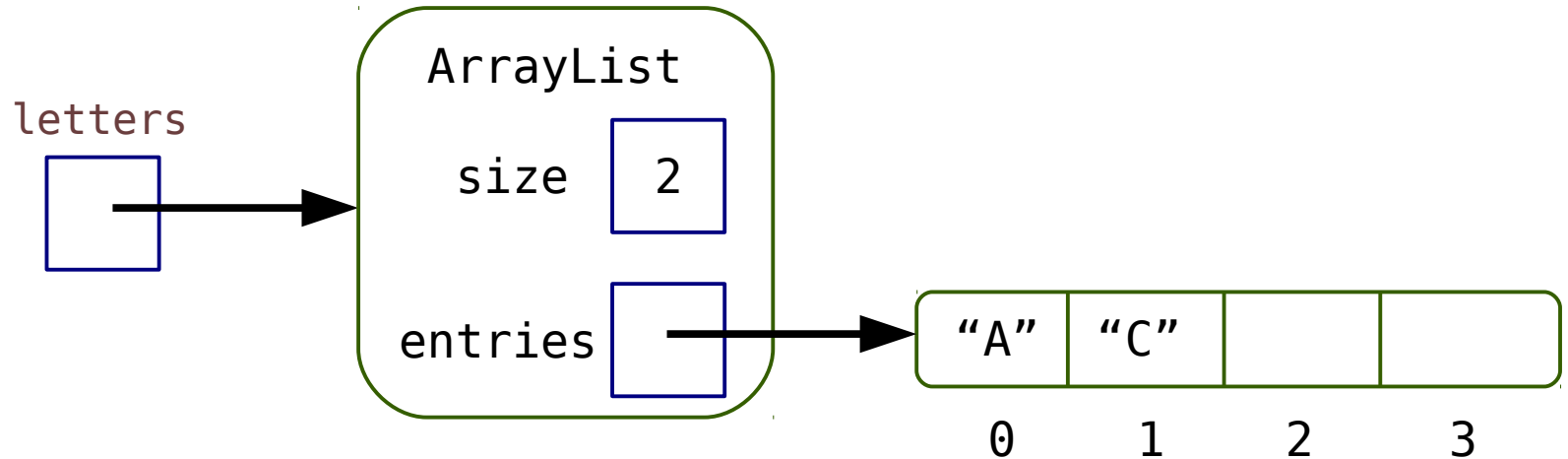- Assume that array size is doubled when needed.

```
ArrayList<String> letters = new ArrayList<>();
letters.add("A");
letters.add(0, "B");
letters.add(1, "C");
letters.remove(0);
```

First line finished for you:

letters

ArrayList

size 0

entries → null | null

        0   1
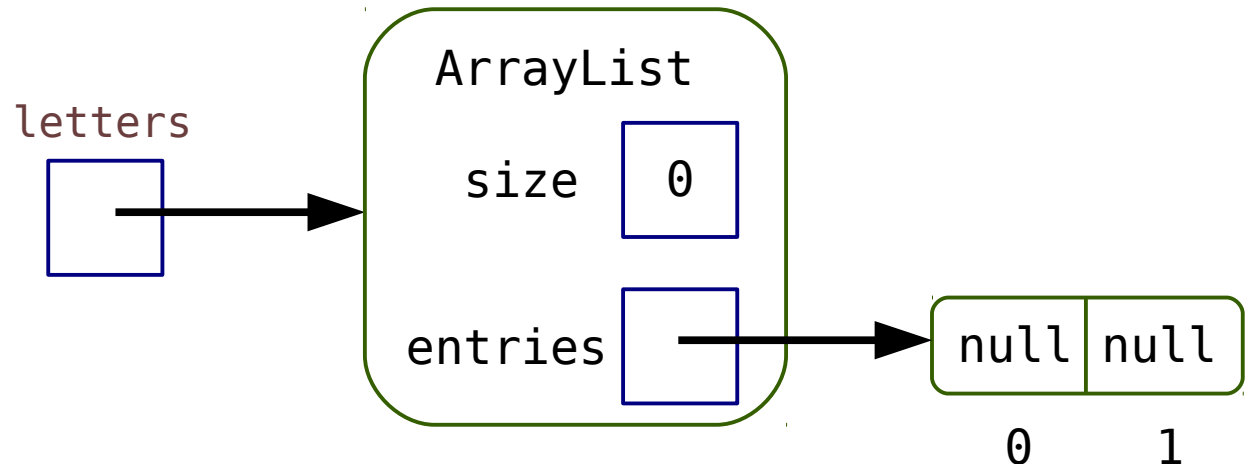
Exercise...

```
ArrayList<String> letters = new ArrayList<>();
letters.add("A");
letters.add(0, "B");
letters.add(1, "C");
letters.remove(0);
```

letters

ArrayList

size   2

entries

| "A"	"C"		
0	1	2	3

ArrayList Weaknesses

- Insertion and deletion near the beginning is slooow
  - Insertion:
    - Every element to the right needs to be shifted right to make space
  - Deletion:
    - Every element to the right needs to be shifted left to fill the gap
- Maybe a LinkedList (??) will do better. Let's try...
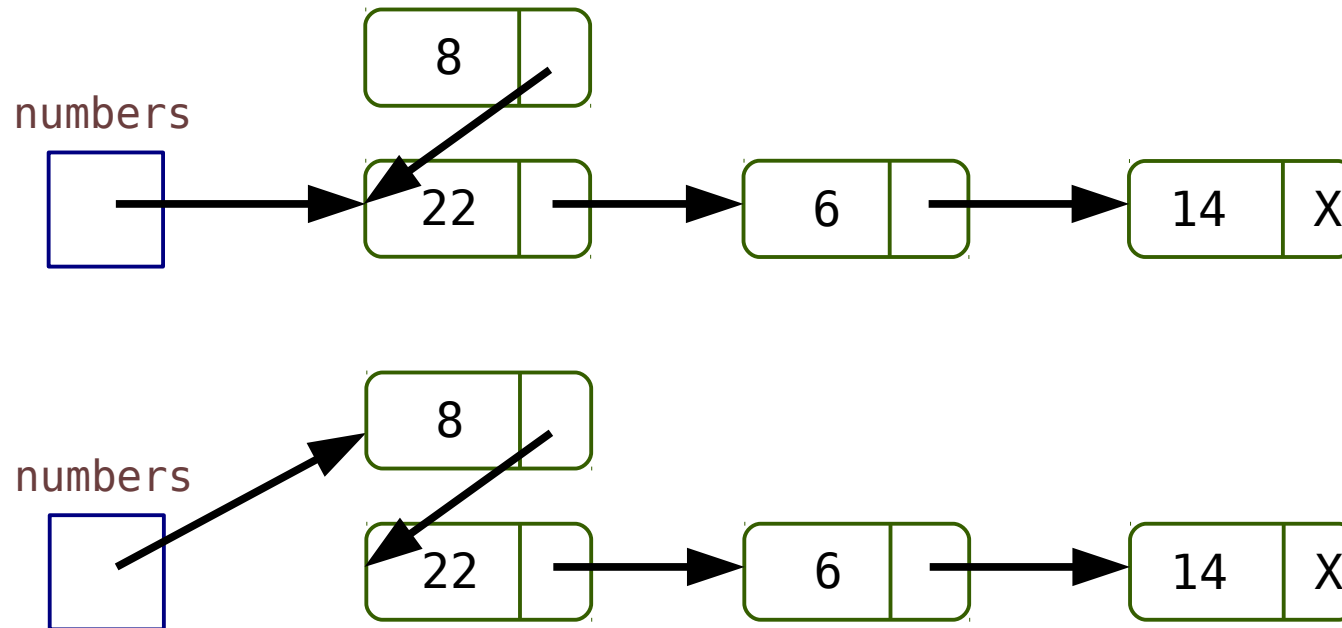
Linked Structures

- Linked structures "chain" elements using references:

numbers
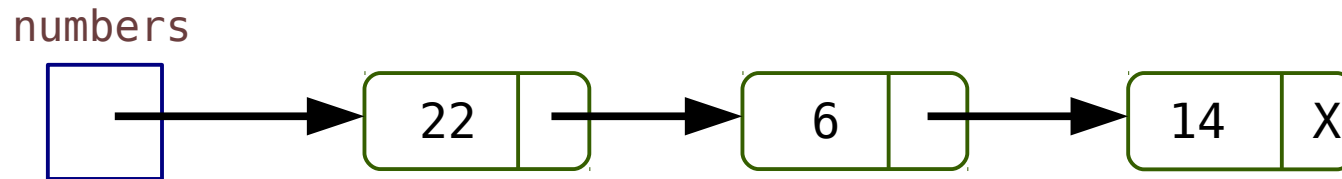
| | → | 22 | | → | 6 | | → | 14 | X |
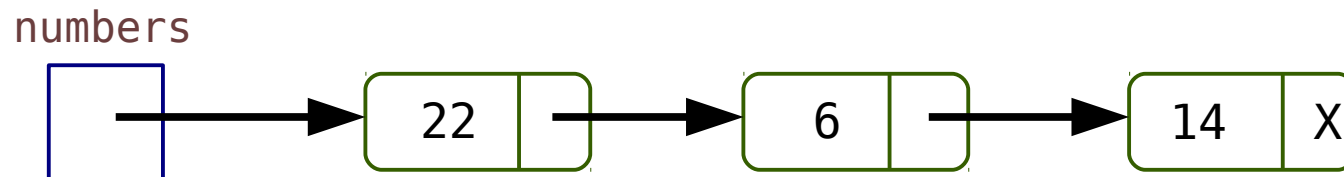
Linked Insertion

- This organization allows fast insertions/deletions near the beginning.
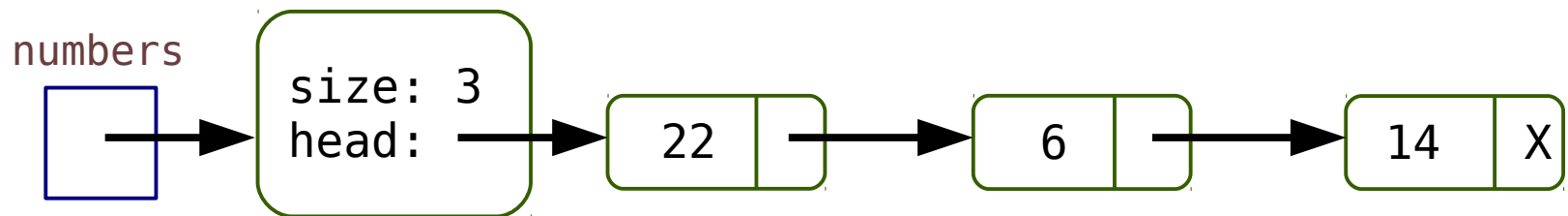
- Adding 8:

Linked List Implementation

- Linked objects are referred to as "nodes"

numbers

| | | 22 | | 6 | | 14 | X |

- Node.java
- NodeDriver.java

LinkedList Implementation

- Inconvenient to work with "naked" nodes.
- Create a wrapper class to handle list logic:

numbers

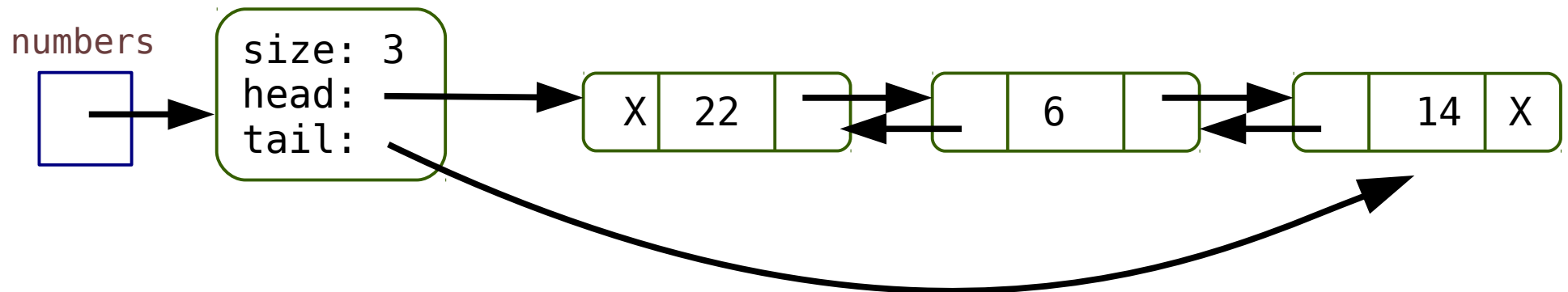| size: 3 head: | → | 22 | → | 6 | → | 14 | X |

- SimpleLinkedList.java
- ListDriver.java

LinkedList Weaknesses

- Can you think of situations where ArrayLists are preferable to LinkedLists?

- ArraysAreGood.java
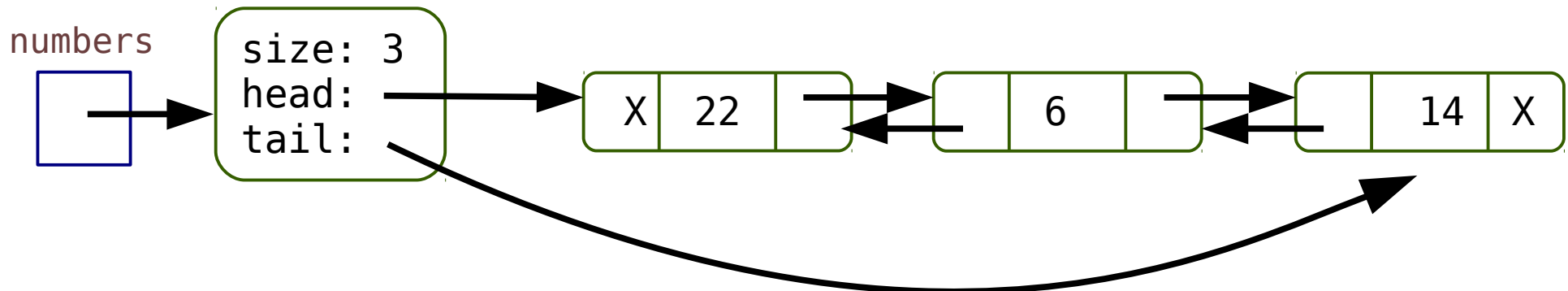
Doubly-Linked Lists

- Java's LinkedList class is doubly-linked:

- What are the advantages to this approach?
- Are there any disadvantages?

Doubly-Linked Lists

- Java's LinkedList class is doubly-linked:


```
numbers          size: 3
                 head:        X  22         6        14  X
                 tail:
```

- What are the advantages to this approach?
  - Efficiently add/remove near either end
  - Efficient backward iteration
- Are there any disadvantages?
  - The backward references require extra space