List Implementations that Use Arrays

Chapter 13

Contents

- Using an Array to Implement the ADT List
  - An Analogy
  - The Java Implementation
  - The Efficiency of Using an Array to Implement the ADT List
- Using a Vector to Implement the ADT List

Objectives

- Implement ADT list by using either array that you can resize or instance of **`Vector`**
- Discuss advantages, disadvantages of implementations presented

Alternatives

- Use an array
  - When all space used, must move data to larger array
- Use Java class **`Vector`**
  - Like an array that can expand automatically
- Chain of linked nodes
  - Insertion/deletion anywhere is harder

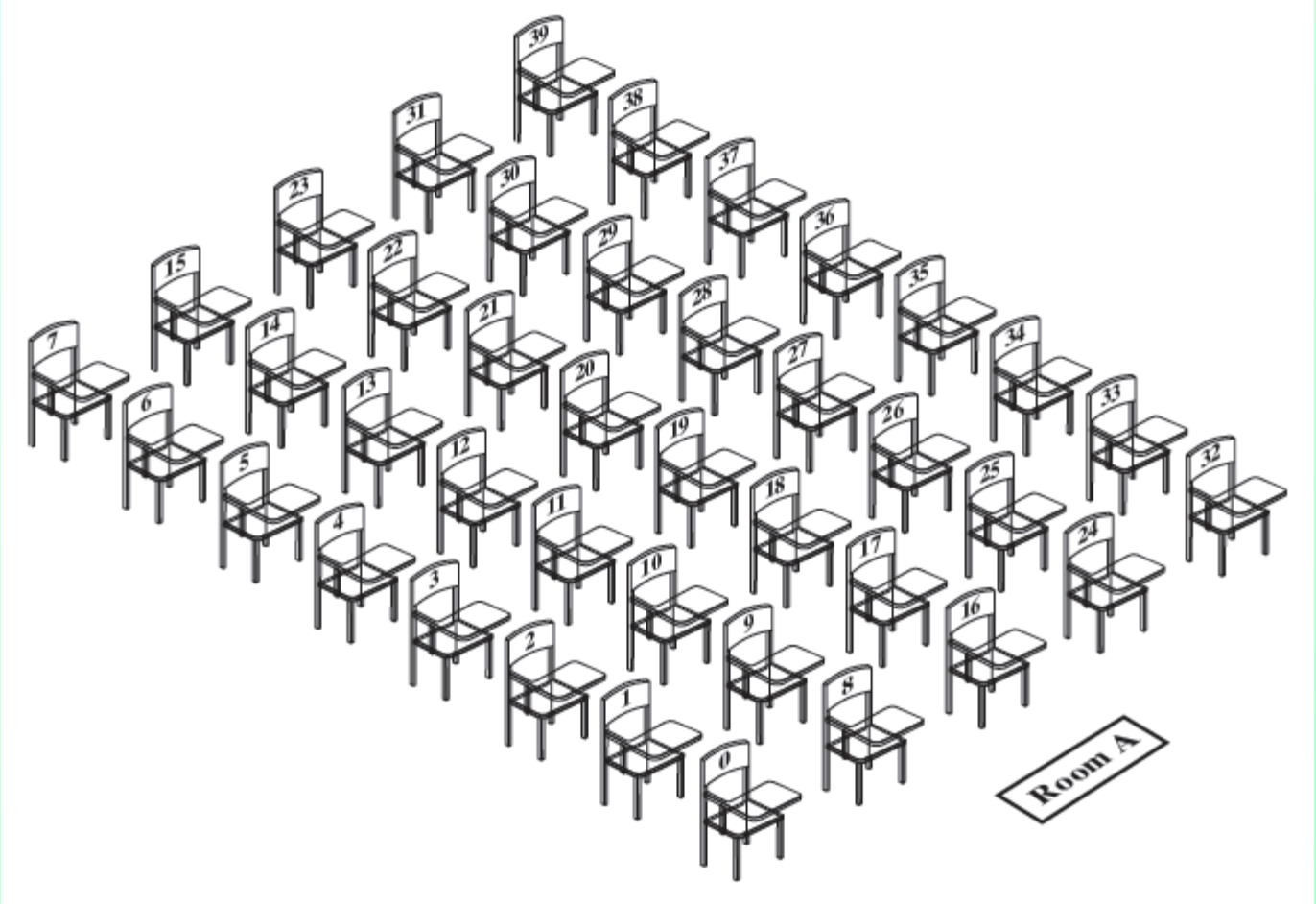

Figure 13-1 A classroom that contains desks in fixed positions

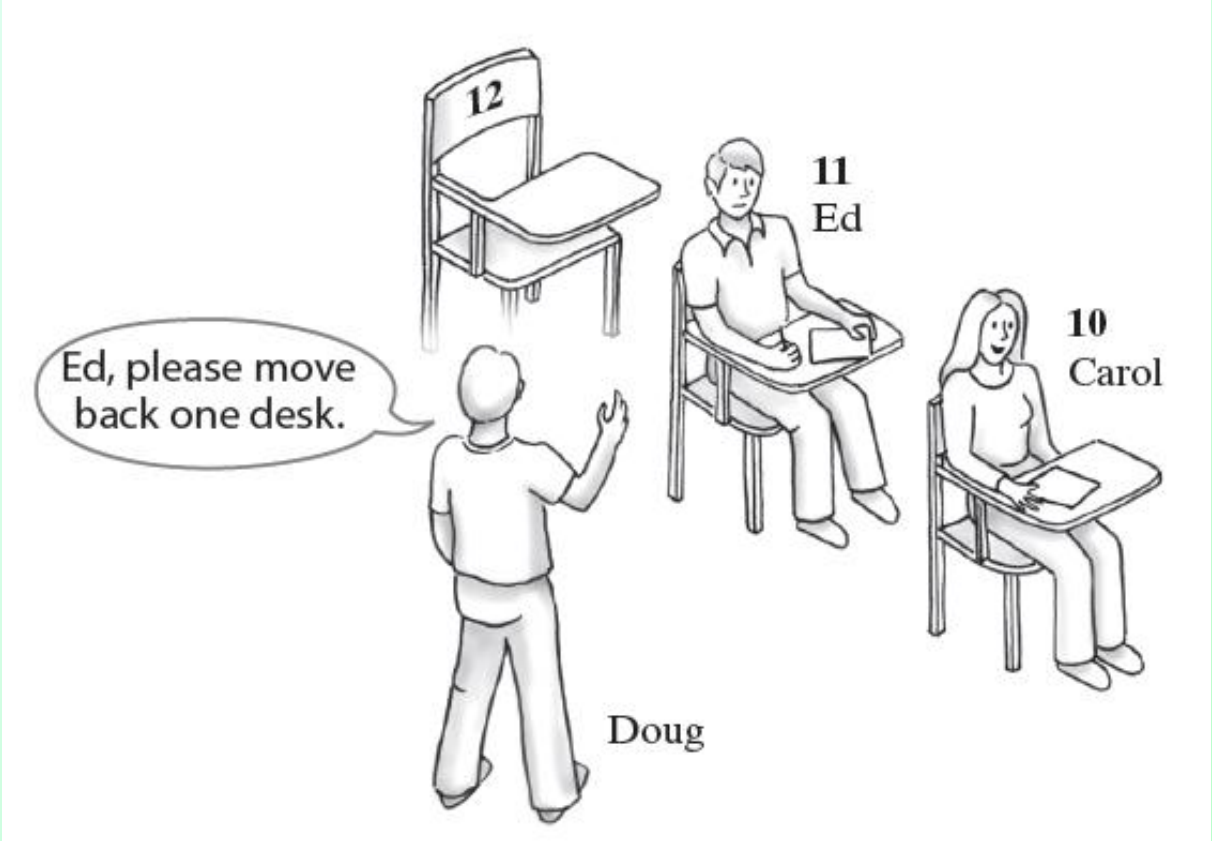

Figure 13-2 Seating a new student between two existing students: At least one other student must move

Question 1 In the previous example, under what circumstance could you add a new student alphabetically by name without moving any other student?

Question 1 In the previous example, under what circumstance could you add a new student alphabetically by name without moving any other student?

When the name comes after the name of the student in the last occupied desk; the new student then sits at the desk after the last one that is currently occupied.

The Java Implementation

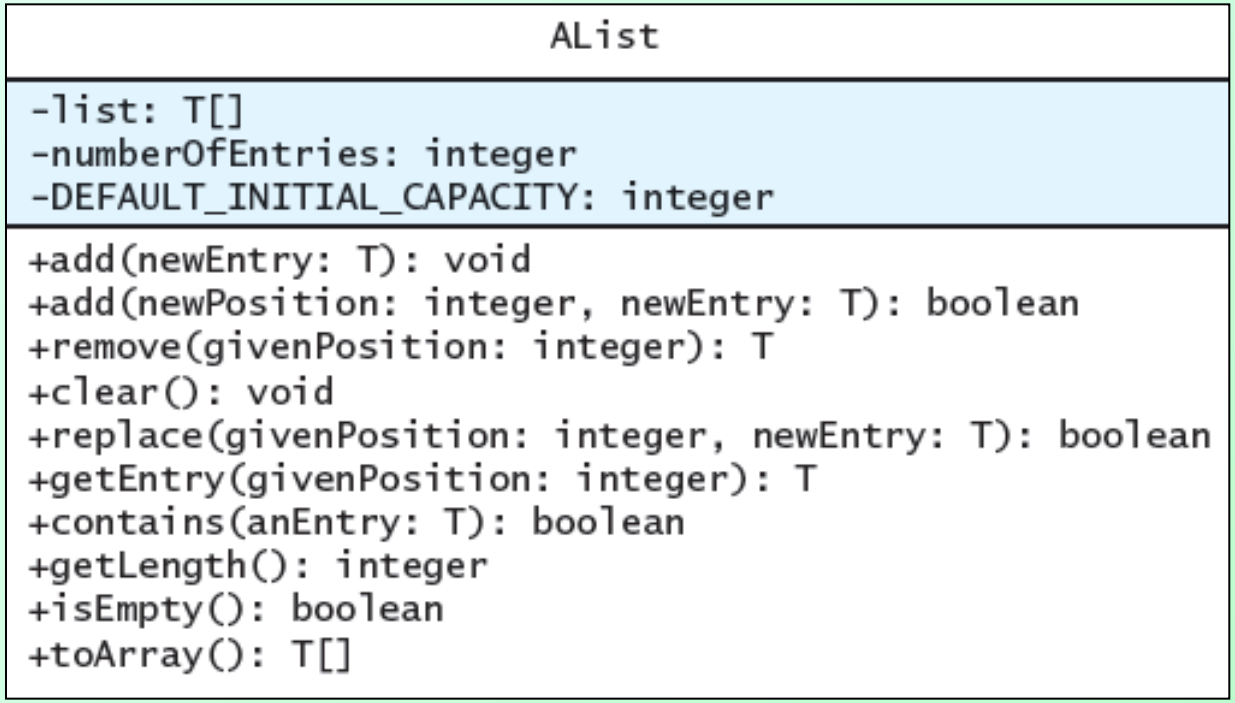

Figure 13-3 UML notation for the class **`AList`**

The Java Implementation

- Note **`Alist`** code, Listing 13-1

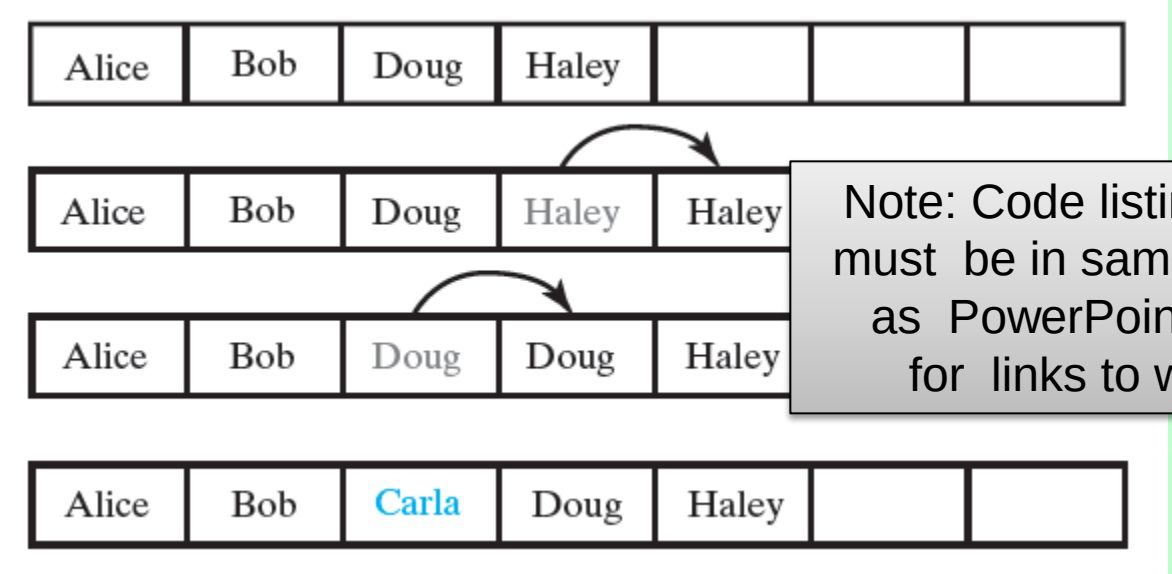

Figure 13-4 Making room to insert Carla as the third entry in an array

Question 4 You could implement the first add method, which adds an entry to the end of the list, by invoking the second add method, as follows:

```
public void add(T newEntry)
{   add(numberOfEntries + 1, newEntry);
}
```

Discuss the pros and cons of this revised approach.

Question 5 Suppose that myList is a list that contains the five entries a b c d e.

a. What does myList contain after executing myList.add(5, w) ?

b. Starting with the original five entries, what does myList contain after executing myList.add(6, w) ?

c. Which of the operations in Parts a and b of this question require entries in the array to shift?

Question 4 You could implement the first add method, which adds an entry to the end of the list, by invoking the second add method, as follows:

```
public void add(T newEntry)
{   add(numberOfEntries + 1, newEntry);
}
```

Discuss the pros and cons of this revised approach.

Advantage: It is easier to implement this add method. Your code will more likely be correct if the other add method is correct.

Disadvantage: Invoking another method uses more execution time. Additionally, the second add method invokes makeRoom needlessly.

Question 5 Suppose that myList is a list that contains the five entries a b c d e.

a. What does myList contain after executing myList.add(5, w) ?

a b c d w e

b. Starting with the original five entries, what does myList contain after executing myList.add(6, w) ?

a b c d e w

c. Which of the operations in Parts a and b of this question require entries in the array to shift?

The operation in Part a

Question 6 If myList is a list of five entries, each of the following statements adds a new entry to the end of the list:

```
myList.add(newEntry);
myList.add(6, newEntry);
```

Which way requires fewer operations?

Question 6 If myList is a list of five entries, each of the following statements adds a new entry to the end of the list:

```
myList.add(newEntry);
myList.add(6, newEntry);
```

Which way requires fewer operations?

myList.add(newEntry). The other add method validates the position 6 and then needlessly invokes makeRoom.

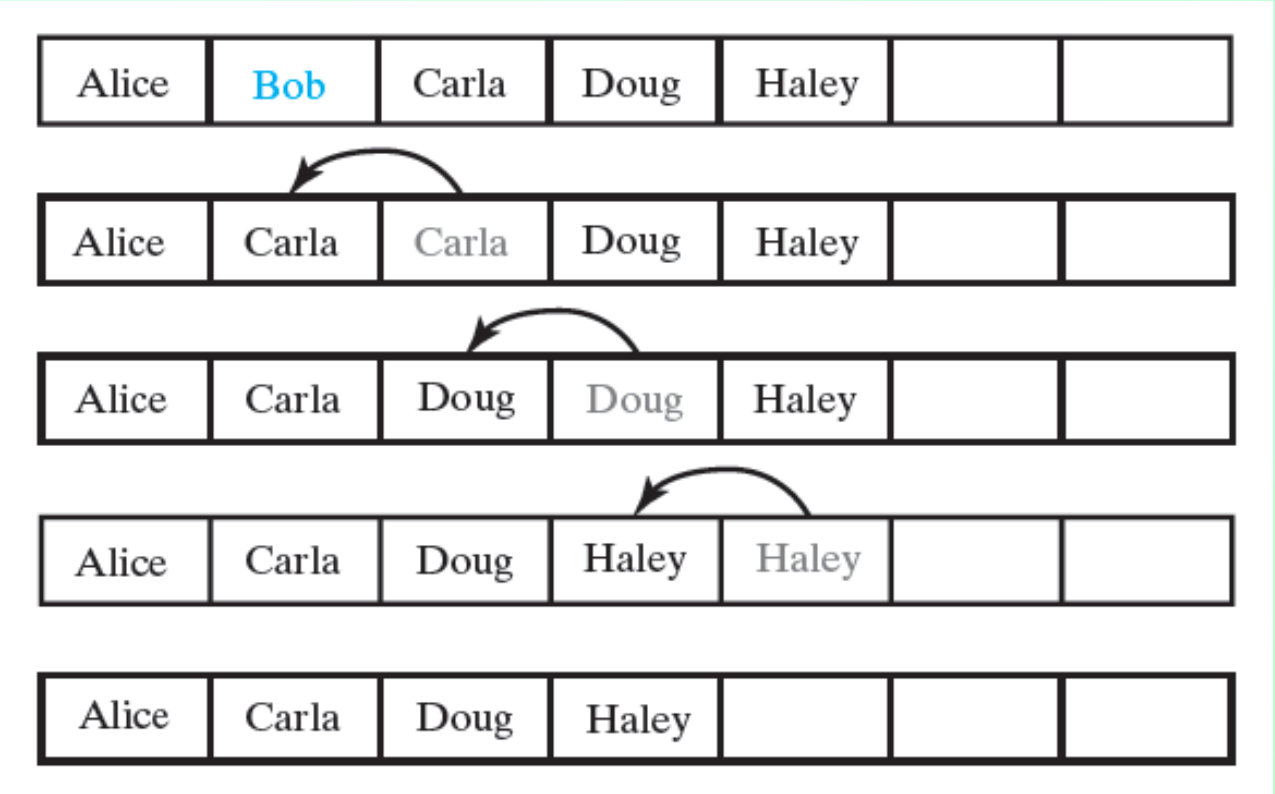

Figure 13-5 Removing Bob by shifting array entries

Using a Vector to Implement the ADT List

- View class **`VectorList`**, Listing 13-A
- Note
  - Example of an adaptor class
  - Writing code for the class simple
  - Execution may be slow due to background invocation of **`Vector`** methods
  - Adding at end of list, retrieving specific entry are fast
  - Adding, removing in middle of list slower

End

Chapter 13