

Linked Lists

Lecturer: Arran Stewart

Outline

- ▶ Why do we use linked lists?
- ▶ How can we implement linked lists?
- ▶ How can we use linked lists to implement other data structures?

Storing lists of values

- ▶ We often need to represent lists of values in our programs – like names of students in a class or a list of things to do.
- ▶ In Java, one way of storing lists of values is to use *arrays*.
- ▶ If you need to access or modify the i th element of an array, computers can do this very quickly.

Disadvantages of arrays

Using arrays to store lists of values comes with some problems:

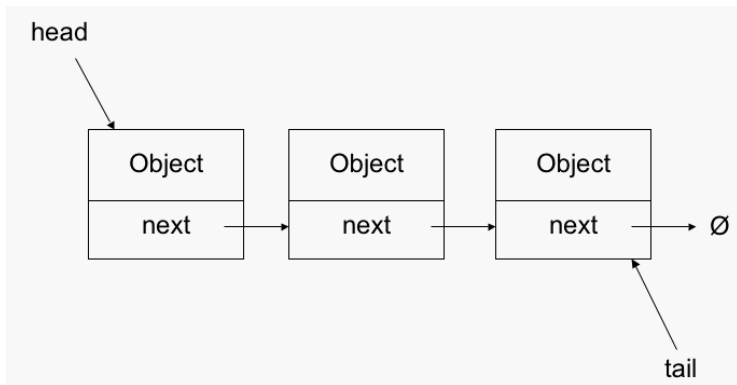
- ▶ When we declare the array, we have to specify up front how many elements it contains.
 - ▶ If we declare a variable using the code `int arr[10]`, then it can store 10 `int` elements.
- ▶ The size can't be changed after the array is declared – if we want to store more elements, we must construct a new array.
 - ▶ This is because Java stores an array as a *contiguous* chunk or “block” of memory – all the elements lie next to one another.
 - ▶ If there are other variables in memory next to the array – it can't be “expanded”, because there's no room for it to do so.

Linked list

- ▶ Sometimes we would like a data structure that is more flexible than an array, even if it is not quite as fast.
- ▶ In this case, we may use a *linked list*.
 - ▶ A linked list consists of a sequence of *nodes*, where each node contains an element. But each node also contains a “link” to the next node in the sequence – hence the name “linked list”.

Diagram of a linked list

We will discuss the following diagram which shows how a linked list is organized.



Java code for a linked list

We will discuss the following partial Java code for a very simple linked list of integers:

```
public class ListNode {  
    //fields (attributes)  
    public int value;  
    public ListNode next;  
  
    //constructor  
    public ListNode(int val) {value = val; next = null; }  
}
```

Advantages of linked lists

- ▶ Unlike with an array, elements of a linked list need not sit right next to each other in memory.
 - ▶ Two nodes that represent consecutive elements of the *list* could be very far away from each other in memory.
- ▶ We don't have to specify up front how many elements a linked list will contain.
 - ▶ As long as the computer has enough memory to create new nodes, we can keep adding elements to the list.

Example linked list code

We will discuss how to create a list of things to do, using the code in

- ▶ `StringListNode.java`
- ▶ `TODOList.java`