【DS】Day4

Tags	
Date	@May 25, 2022
Summary	Linked Stack and Array Stack

【Week2】Stacks

2.1 Stacks

Operations: Insert, remove, iterate, test if empty

Stack API

```
public class StackOfStrings {
  StackOfStrings() // Create an empty stack
  void push(String item) // Insert a new string onto the stack
  String pop() // Remove and return the string most recently added
  boolean isEmpty() // Is the stack empty
  int size() // Number of elements on the stack
}
```

Stack Test Client

```
public static void main(String[] args) {
  StackOfStrings stack = new StackOfStrings();
  while (!StdIn.isEmpty()) {
    String s = StdIn.readString();
    if (s.equals("-"))
      StdOut.println(stack.pop());
    else
      stack.push(s);
  }
}
```

Pop

```
String pop() {
  String item = first.item;
  first = first.next;
  return item;
}
```

Push

```
void push(String item) {
  Node oldFirst = first;
  // Create a new node for the beginning
  first = new Node();
  // Set the intance variables in the new node
  first.item = item;
  first.next = oldFirst;
}
```

Array Implementation of A Stack

- Use array `s[]` to store N items on stack
- `push()`: add new item at s[N]
- `pop()`: remove item from `s[N-1]`

```
public class ArrayStack {
  String stack[];
  int N = 0;

  public ArrayStack(int capacity) {
    stack = new String[capacity];
  }

  public void push(String item) {
    stack[N++] = item;
  }

  public String pop() {
    String item = stack[--N];
    stack[N] = null;
    return item;
  }
  public boolean isEmpty() {
    return N == 0;
  }
}
```