CSC 1052 – Algorithms & Data Structures II: Linked Stack

Professor Henry Carter
Spring 2017

Recap

- Linked list allow data to be stored in random locations in memory
- Typically require more time to access in exchange for more efficient memory usage
- Implementation pitfalls frequently related to pointers
- Variants allow for more flexible access at the cost of method complexity

Implementations

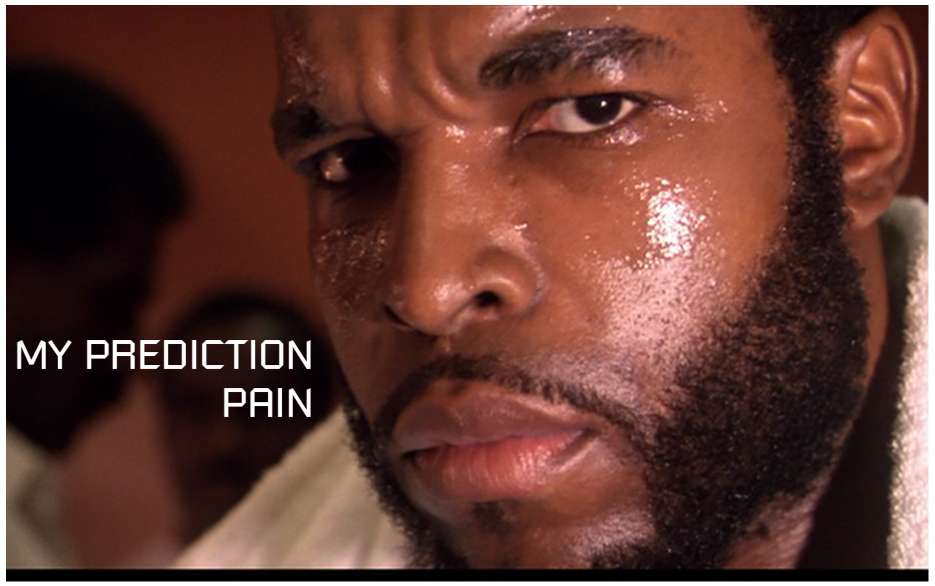
- May data structures we consider will have both array and linked list implementations
- Comparison between implementations will be a critical learning goal
- Some of these data structures are implemented already
  - Example: Java ArrayLists vs LinkedLists (we'll cover the abstract list description later this semester)

Stack Implementations

- ArrayBoundedStack:
  - Fast access for stack operations
  - Limited by array size
- ArrayList Stack
  - Simple layered approach to implementation
  - Built on arrays with re-sizing automated inside ArrayList (but still requiring linear access time)

Linked List Stack

- StackInterface requires:
  - Push
  - Pop
  - Top
  - isEmpty
  - isFull
- Which operations will be faster?
- Which operations will be slower?



LinkedStack Class

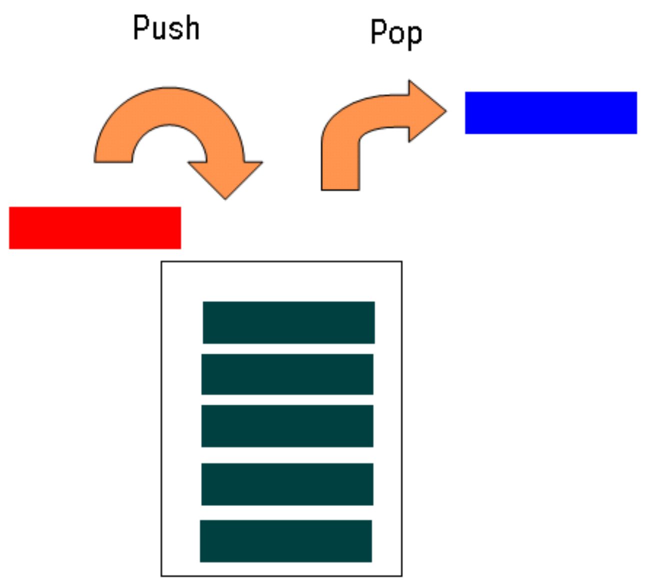
- Class variables
- Constructor
- Interface methods

Comparison: Order of Growth

- Push
- Pop
- Top
- isEmpty, isFull
- Hidden costs: accessing memory vs allocating memory

Java Stacks

- Implements the Stack ADT on an array
    - Buried inside another abstraction called "Vector"
- Use generics to store objects of any type
- Part of java.util.*

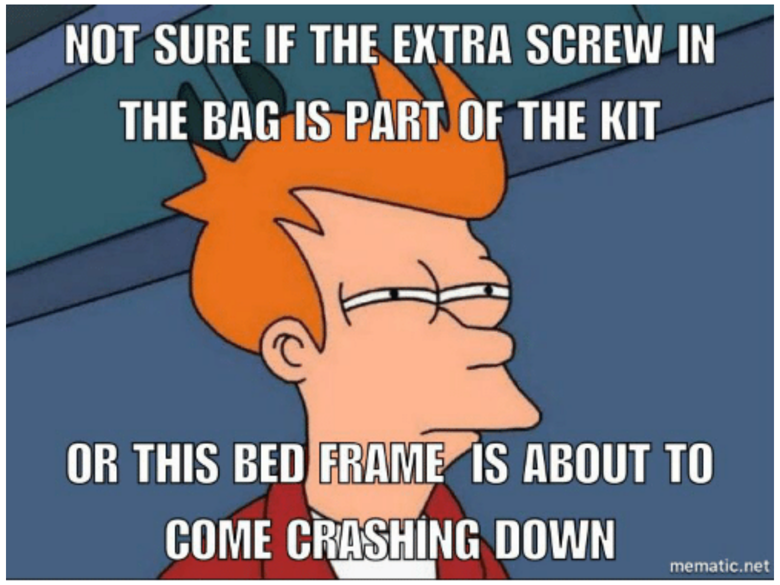


Java Stack Methods

- Push
- Pop
  - Both removes and returns
- Peek == Top
- Search()
  - Returns the index of the input object

Extra Methods?

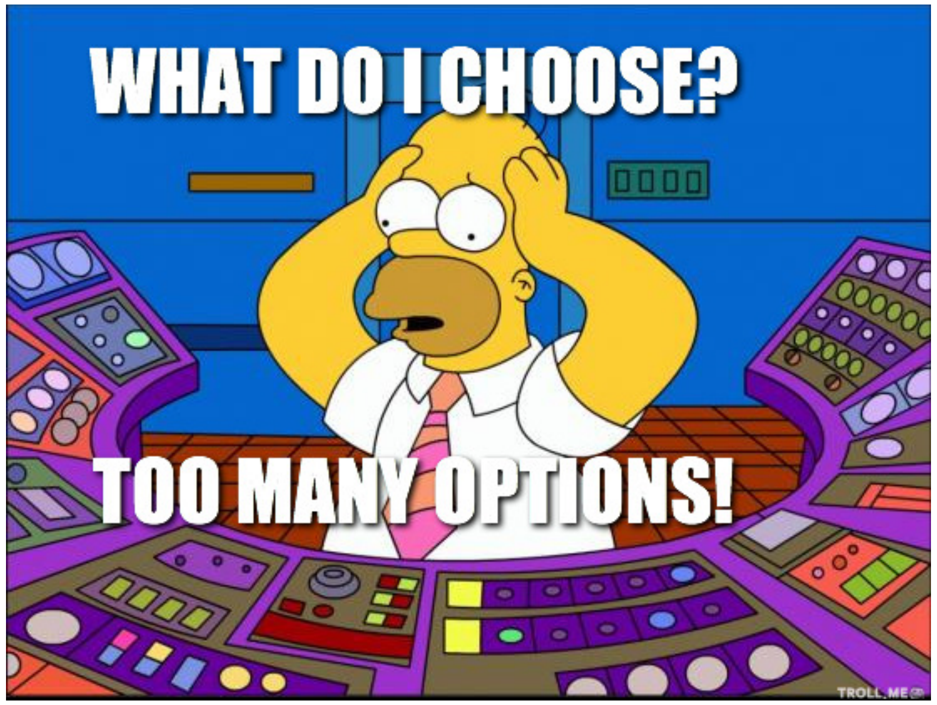
- The Stack (and other ADTs we will study) are implemented already in the Java collections framework
- Many of these ADTs extend other ADTs, so they have additional functionality beyond the canonical ADT functions
- Be careful about abusing this behavior!


NOT SURE IF THE EXTRA SCREW IN THE BAG IS PART OF THE KIT
OR THIS BED FRAME IS ABOUT TO COME CRASHING DOWN
mematic.net

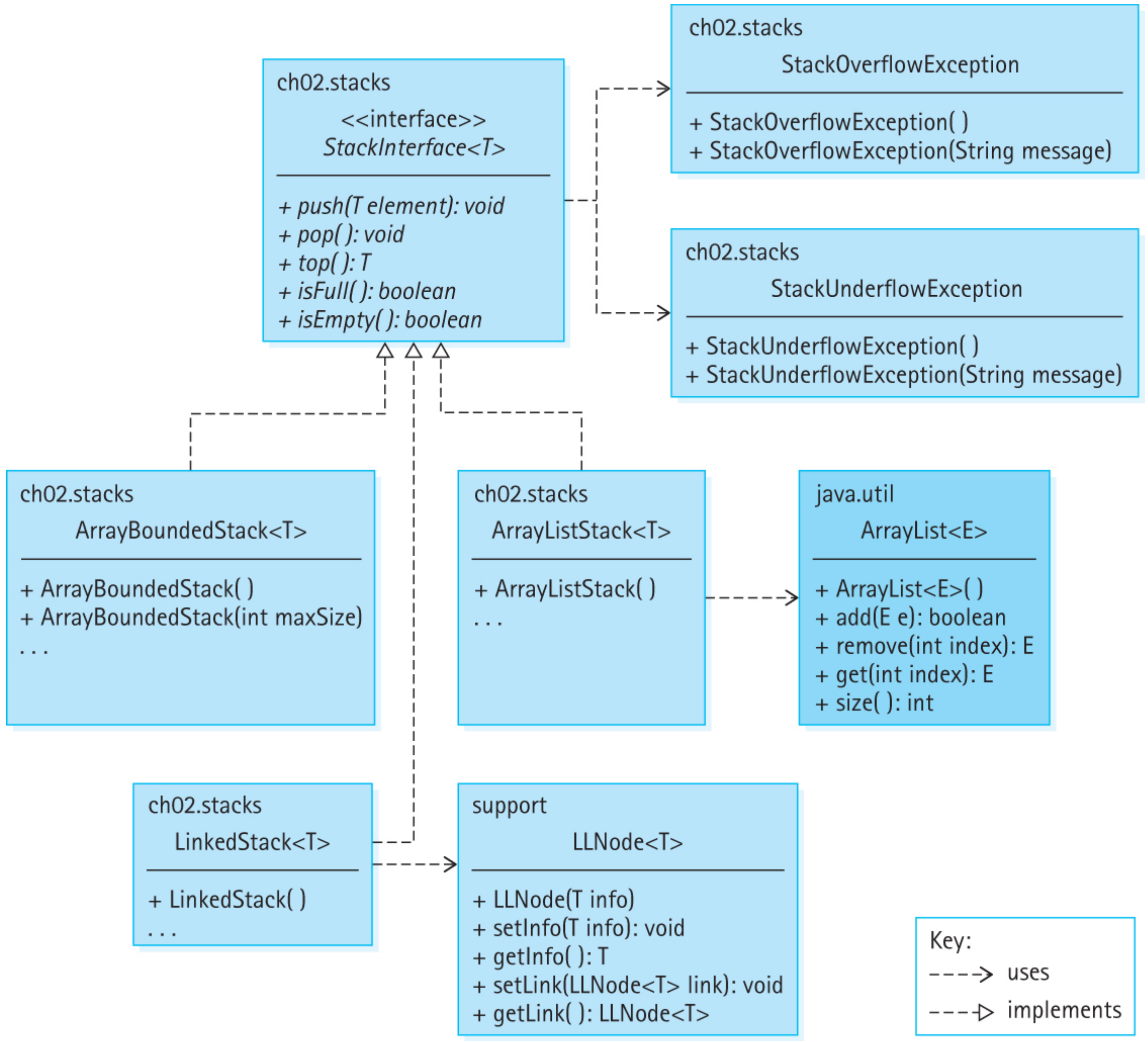
Building IKEA furniture at 2 am might not have been the best idea

Multiple Interfaces

- Navigate to the Stack class on the java API
- What class does it extend?
- What interface(s) does it implement?



Chapter 2 UML Overview



Exercises

- Do we need a top() method?
- How many classes implement the Collection<T> interface?
- Determine what the code snippet does:

```
String b = scan.next();
Stack<Character> a = new Stack<>();
for(int i = 0;i < b.length();i++){
    a.push(b.charAt(i));
}

boolean c = true;
for(int i = 0;i < b.length();i++){
    if(b.charAt(i) != a.pop()){
        c = false;
        break;
    }
}
System.out.println(c);
```

Recap

- The Stack ADT can be efficiently implemented as a linked list
- The pros of the linked list implementation are most notable when the array stack has to be re-sized a lot
- The Java collections framework provides implementations of many of the ADTs we will be studying

Next Time...

- Dale, Joyce, Weems Chapter 3.1-3.2
  - Remember, you need to read it BEFORE you come to class!
- Check the course webpage for practice problems
- Peer Tutors
  - http://www.csc.villanova.edu/help/