Lecture 13: Stacks
Reading about Collection Classes

- Oracle’s Java Tutorials
  - Trail: Collections
  - https://docs.oracle.com/javase/tutorial/collections/
Stack

• Interface Stack<E> {
  • void push(E value)
  • E pop()
  • E peek()
}

• Example: Trays in cafeteria

• Last In - First Out (LIFO)
  No changes to middle of list ever!
Stack Applications

- Run-time stack:
  - See sum program
- Backtracking
  - Solving Maze
- Evaluating expression in postfix form:
  - $(52 - ((5 + 7) * 4)) \Rightarrow 52 \ 5 \ 7 \ + \ 4 \ * \ - \ \Rightarrow \ 4$
- Tools to parse programs
- Undo
Stack Implementations

• ArrayList:
  • Which end should be head?
  • How complex for push, pop, peek?

• SinglyLinkedList:
  • Which end should be head?
  • How complex for push, pop, peek?

• Space differences?
  • What if there are several stacks?

• java.util.Stack based on Vector - don’t use!
  • ArrayDeque is better choice (more details later)
Queue

• FIFO: Waiting in line
• Operations:
  • enqueue (at end) – or add
  • dequeue (from beginning) – or remove
• Examples:
  • Simulations
  • Event queue
  • Keeping track when searching
Queue Implementations

- **SinglyLinkedList:**
  - Which end should be front, rear?
  - How complex for enqueue, dequeue?

- **ArrayList:**
  - Which end should be front, rear?
  - How complex for enqueue, dequeue?

- **Space differences?**