Stacks

- Organizing data
- Stack operations and interface
- Stack Class implementation
- Examples and applications
- Stack Class in Java

Organization of Data

- Organize data by position
  - stacks
  - queues
- Organize data by value
  - lists (arrays, vectors, linked lists)
  - trees
  - priority queues

Stacks as Data Structures

- To simulate the behavior of phenomena where insertions and deletions occur at the same end of the data structure
  - pile of books
  - cafeteria trays
  - laundry basket
- LIFO -- last item in is the first out
- Only the top element is relevant!

Stack Operations

- initialize the stack
- is the stack empty?
- is the stack full?
- inserting an item
- deleting an item
- retrieving an item

Stack Interface

- boolean isEmpty()
- void push(E obj)
  - inserts object at the top of the stack
- E pop()
  - deletes the top of the stack, returns an object
- E peek()
  - retrieves the top of the stack, returns an object

Stack Implementation

- Array-based (arrays)
  - pros and cons
  - I wrote my own implementation *ArrayStack.java*
- Reference-based (linked list)
  - pros and cons
  - I wrote my own implementation *LinkedStack.java*

Stack runtime analysis

- Using an index-based implementation
- push() may require increasing size of the array
- pop() may require decreasing size of the array

	best	worst	amortized
push	O(1)	O(n)	O(1)
pop	O(1)	O(n)	O(1)
size	O(1)	O(n)	O(1)

Decrease size of array when usage is less than 25% of the capacity. How?

Stacks Applications

- Memory management
  - activation of records
  - stack frame
  - recursion
- Evaluation/conversion of expressions in postfix, infix, and prefix notations
- Backtracking algorithm (search)
- Recognizing strings in a language

Nested parentheses

- Determine if a string containing a mathematical expression is nested correctly.
  - Parenthetical symbols include { } , [ ], ( )

- Algorithm
  - Process the expression from left to right.
  - Push each left parenthetical symbol on a stack.
  - When you encounter a right parenthetical symbol, pop the stack to see if there is a matching left parenthetical symbol. If not, the parenthetical nesting is invalid.
  - After the entire expression is processed, if the stack is empty, the parenthetical nesting is valid.

Notation for Expressions

- infix

  (7 - 2) * (3 + 1)

- prefix

  * - 7 2 + 3 1

- postfix

  7 2 - 3 1 + *

Postfix notation (RPN)

- Let each entry in a postfix expression be a *token*.
- Let S be an empty stack.
- For each token in the expression:
  - If the token is a number, push it on stack S.
  - Otherwise (the token is an operator):
    - Pop a token off stack S and store it in y.
    - Pop a token off stack S and store it in x.
    - Evaluate: x operator y.
    - Push result on stack S.
- Pop the stack for the final answer.

Run-time Stack

- Java maintains a *run-time stack during the execution of your* program.
- Each call to a method generates an *activation frame that* includes storage for:
  - arguments to the method
  - local variables for the method
  - return address of the instruction that called the method
- A call to a method pushes an activation record on the run-time stack.
- A return from a method pops an activation record from the run-time stack.

Stack Class in Java

- Java has a predefined Class Stack<E>
- look it up at the API Java web site

Homework

- Quiz #6 on Wednesday (ArrayList, LinkedList runtime analysis)
- Homework #6 (Stacks & Queues) due on 10/27