

Lecture 6

Stacks and Queues

Feb 23, 2017

Pg. 131

Pg. 131

1. `void clear()`

Pg. 131

1. `void clear()`

2. `bool isEmpty()`

Pg. 131

1. `void clear()`
2. `bool isEmpty()`
3. `void push()`

Pg. 131

1. `void clear()`
2. `bool isEmpty()`
3. `void push()`
4. `T* pop()`

Pg. 131

1. `void clear()`
2. `bool isEmpty()`
3. `void push()`
4. `T* pop()`
5. `T* topEl()` (Optional, and usually called `peek()`)

Example Usage

```
1 Stack<int> myStack;  
2 myStack.push(2);  
3 myStack.push(3);  
4 myStack.push(5);  
5 std::cout << *mystack.pop() << std::endl; // 5  
6 std::cout << *mystack.pop() << std::endl; // 3  
7 std::cout << *mystack.pop() << std::endl; // 2
```

Delimiter Matching (pg. 133)

Operations On Large Numbers (pg. 135)

Evaluate Post-Fix Notation

(https://en.wikipedia.org/wiki/Reverse_Polish_notation)

Pg. 139

Pg. 139

1. `void clear()`

Pg. 139

1. `void clear()`

2. `bool isEmpty()`

Pg. 139

1. `void clear()`
2. `bool isEmpty()`
3. `void enqueue()`

Pg. 139

1. void clear()
2. bool isEmpty()
3. void enqueue()
4. T* dequeue()

Pg. 139

1. `void clear()`
2. `bool isEmpty()`
3. `void enqueue()`
4. `T* dequeue()`
5. `T* firstEl()` (Optional, and usually called `peek()`)

Example Usage

```
1 Queue<int> myQueue;  
2 myQueue.enqueue(2);  
3 myQueue.enqueue(3);  
4 myQueue.enqueue(5);  
5 std::cout << *myqueue.dequeue() << std::endl; // 2  
6 std::cout << *myqueue.dequeue() << std::endl; // 3  
7 std::cout << *myqueue.dequeue() << std::endl; // 5
```

In my experience, you are more likely to use a queue than a stack, but somehow it is more difficult to find simple examples.

Networking

Operating Systems

Messaging

We will explore two different implementations of stacks, and three implementations of queues:

We will explore two different implementations of stacks, and three implementations of queues:

1. Linked List (Stack and Queue) (pg. 137, pg. 143)

We will explore two different implementations of stacks, and three implementations of queues:

1. Linked List (Stack and Queue) (pg. 137, pg. 143)
2. Vector (Stack and Queue) (pg. 135, pg. 151)

We will explore two different implementations of stacks, and three implementations of queues:

1. Linked List (Stack and Queue) (pg. 137, pg. 143)
2. Vector (Stack and Queue) (pg. 135, pg. 151)
3. Array (Queue) (pg. 141)