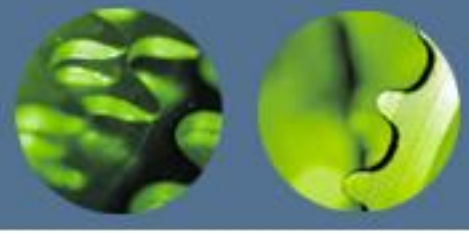
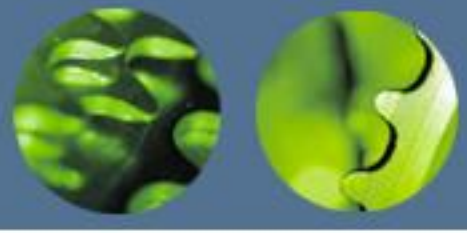
Stacks and Queues



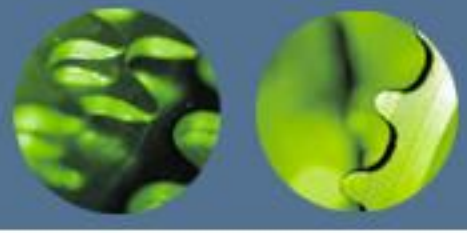
Abstract Data Type

- Abstract Data Type as a design tool
- Concerns only on the important concept or model
- No concern on implementation details.
- Stack & Queue is an example of ADT
- An array is not ADT.



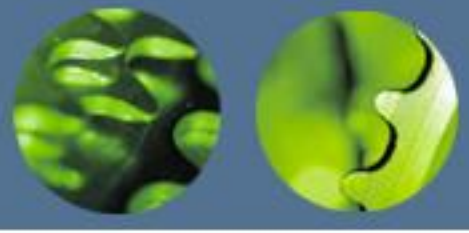
What is the difference?

- Stack & Queue vs. Array
  - Arrays are data storage structures while stacks and queues are specialized DS and used as programmer's tools.
- Stack – a container that allows push and pop
- Queue - a container that allows enqueue and dequeue
- No concern on implementation details.
- In an array any item can be accessed, while in these data structures access is restricted.
- They are more abstract than arrays.



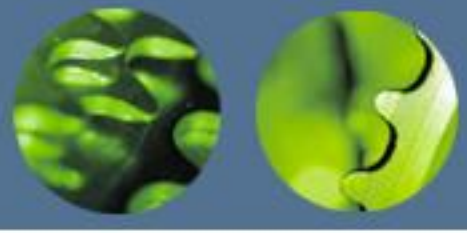
Questions?

- Array is not ADT
- Is Linked list ADT?
- Is Binary-tree ADT?
- Is Hash table ADT?
- What about graph?



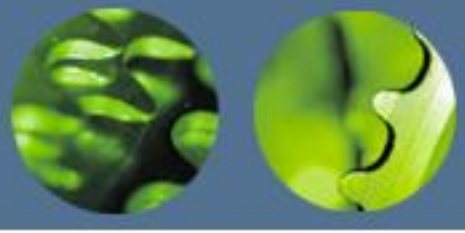
Stacks

- Allows access to only the last item inserted.
- An item is inserted or removed from the stack from one end called the “top” of the stack.
- This mechanism is called Last-In-First-Out (LIFO).



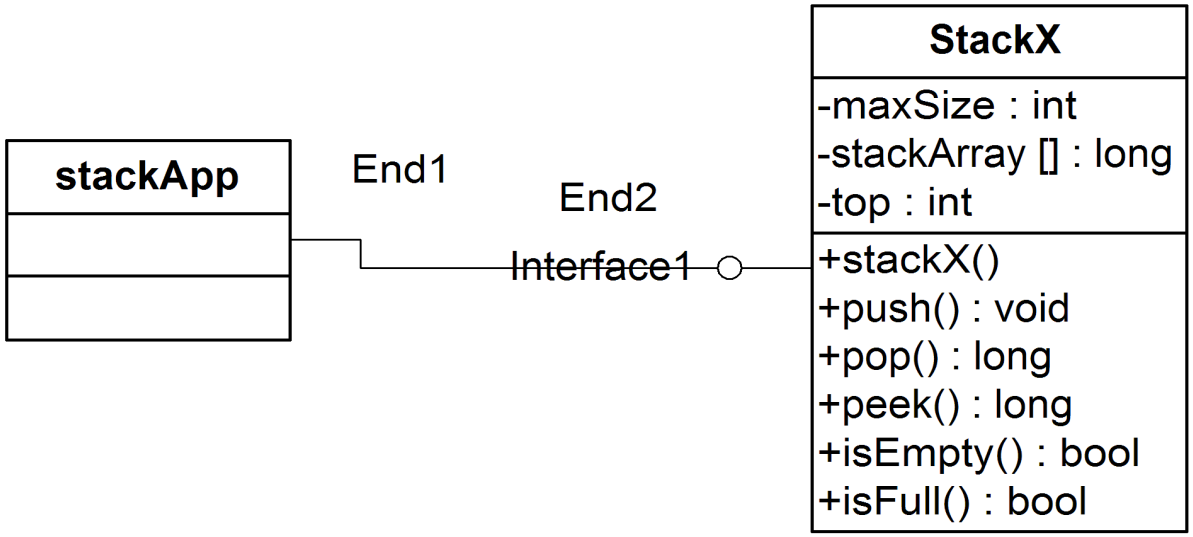
Stack operations

- Placing a data item on the top is called “pushing”, while removing an item from the top is called “popping” it.
- *push* and *pop* are the primary stack operations.
- Some of the applications are : microprocessors, some older calculators etc.

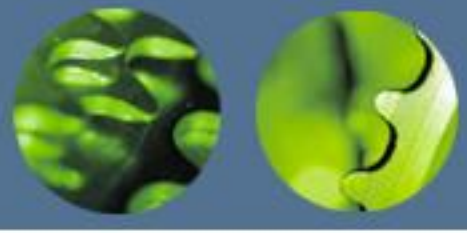


Example of Stack codes

- First example stack ADT and implementation

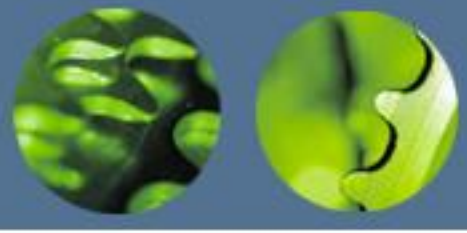


- *push* and *pop* operations are performed in O(1) time.



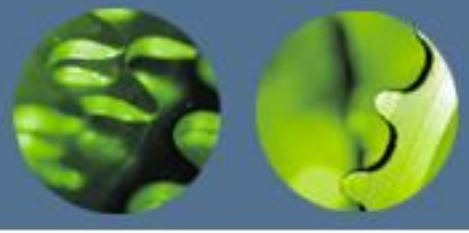
Example of Stack codes

- Reversed word
- What is it?
- ABC -> CBA



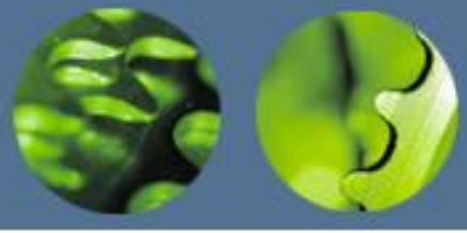
Example of Stack codes

- BracketChecker (balancer)
- A syntax checker (compiler) that understands a language containing any strings with balanced brackets ‘{‘ ‘[‘ ‘(‘ and ‘)’, ‘]’, ‘}’
  - S -> Bl S1 Br
  - S1 -> Bl string Br
  - Bl -> ‘{‘ | ‘[‘ | ‘(‘
  - Br -> ‘)’, | ‘]’, | ‘}’



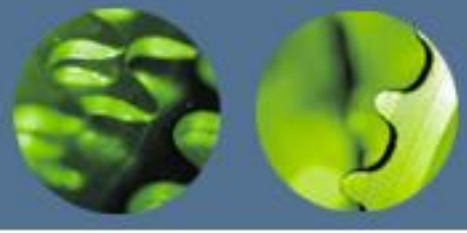
Queues

- Queue is an ADT data structure similar to stack, except that the first item to be inserted is the first one to be removed.
- This mechanism is called First-In-First-Out (FIFO).
- Placing an item in a queue is called “insertion or enqueue”, which is done at the end of the queue called “rear”.
- Removing an item from a queue is called “deletion or dequeue”, which is done at the other end of the queue called “front”.
- Some of the applications are : printer queue, keystroke queue, etc.



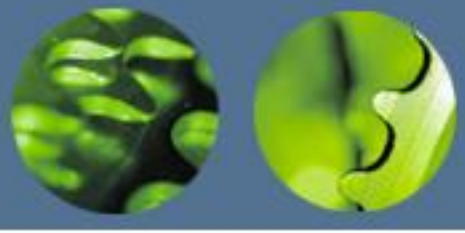
Circular Queue

- When a new item is inserted at the rear, the pointer to rear moves upwards.
- Similarly, when an item is deleted from the queue the front arrow moves downwards.
- After a few insert and delete operations the rear might reach the end of the queue and no more items can be inserted although the items from the front of the queue have been deleted and there is space in the queue.

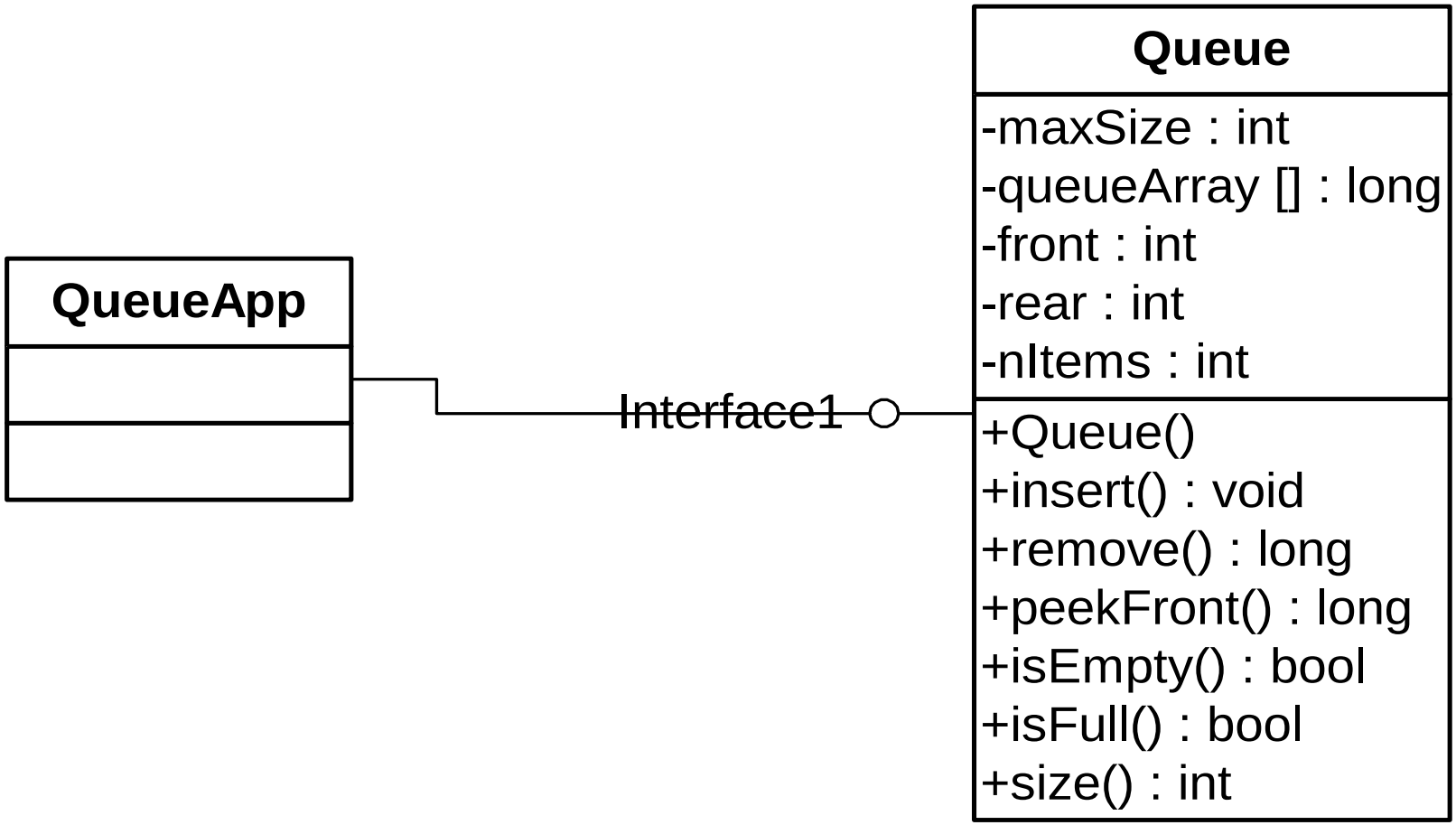


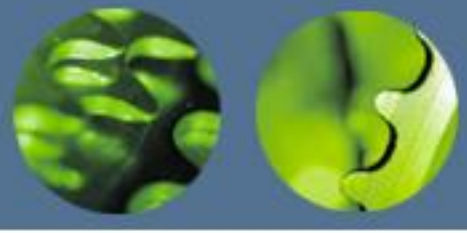
Circular Queue

- To solve this problem, queues implement wrapping around. Such queues are called Circular Queues.
- Both the front and the rear pointers wrap around to the beginning of the array.
- It is also called as “Ring buffer”.
- Items can inserted and deleted from a queue in O(1) time.

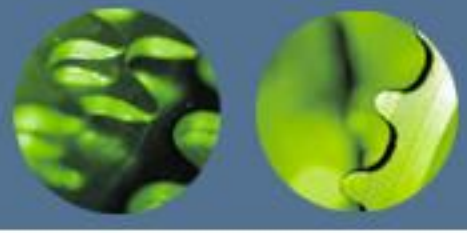


Queue Example



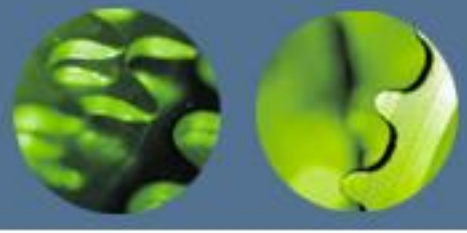


Queue sample code



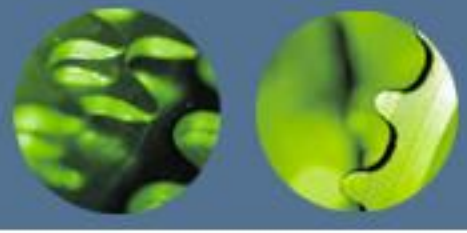
Various Queues

- Normal queue (FIFO)
- Circular Queue (Normal Queue)
- Double-ended Queue (Deque)
- Priority Queue



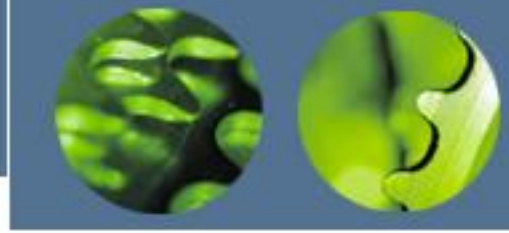
Deque

- It is a double-ended queue.
- Items can be inserted and deleted from either ends.
- More versatile data structure than stack or queue.
- E.g. policy-based application (e.g. low priority go to the end, high go to the front)
- In a case where you want to sort the queue once in a while, What sorting algorithm will you use?

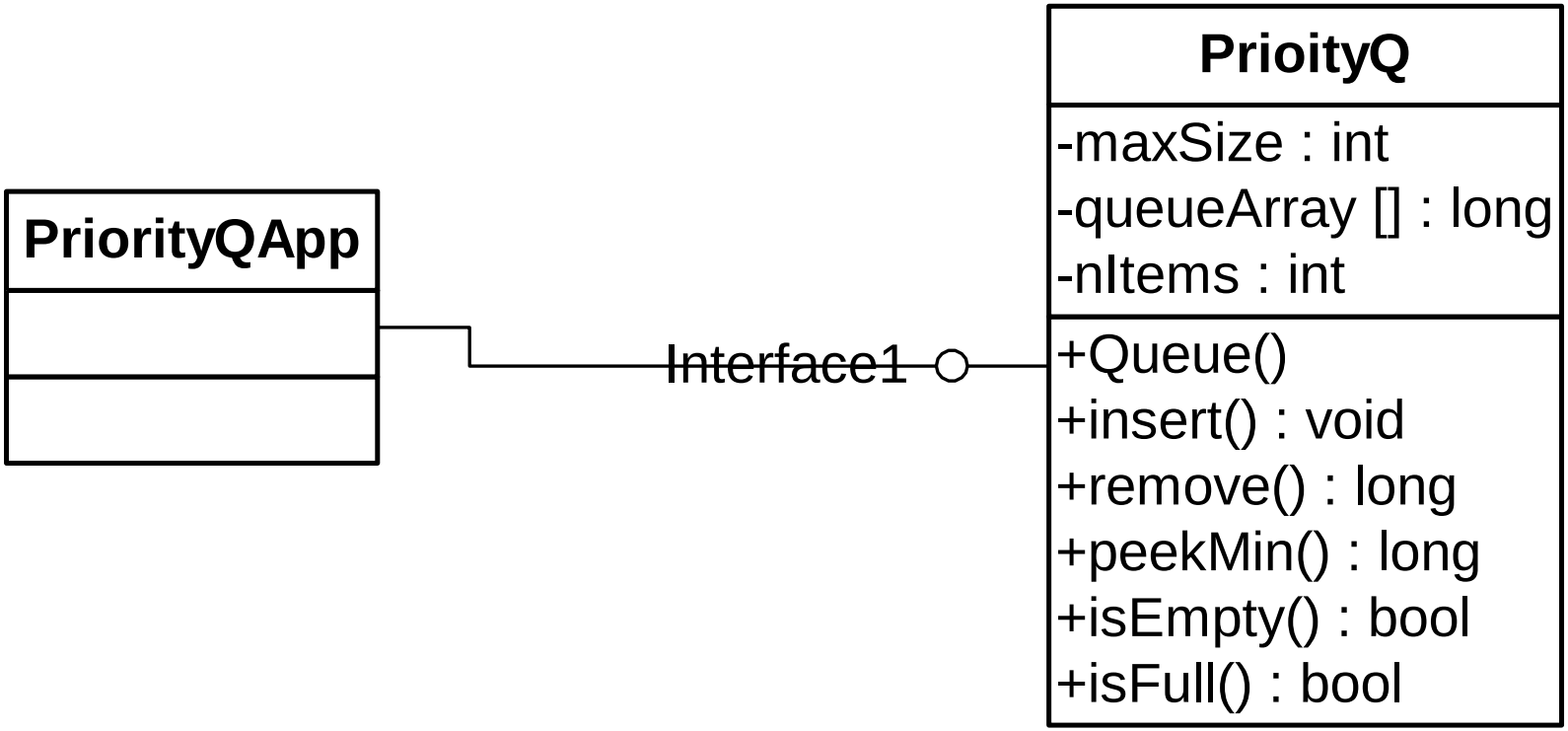


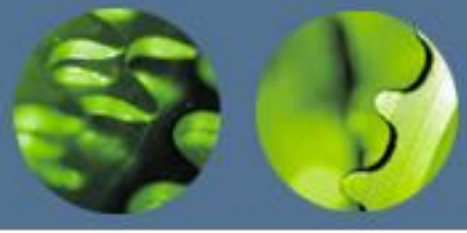
Priority Queues

- More specialized data structure.
- Similar to Queue, having front and rear.
- Items are removed from the front.
- Items are ordered by key value so that the item with the lowest key (or highest) is always at the front.
- Items are inserted in proper position to maintain the order.
- Let’s discuss complexity



Priority Queue Example





Priority Queues

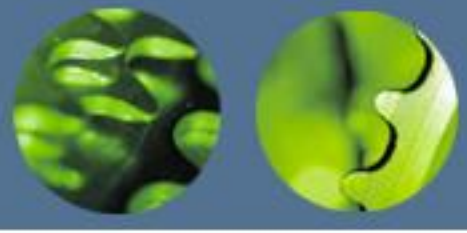
- Used in multitasking operating system.
- They are generally represented using “heap” data structure.
- Insertion runs in O(n) time, deletion in O(1) time.

Parsing Arithmetic Expressions

- 2 + 3
- 2 + 4 * 5
- ((2 + 4) * 7) + 3* (9 – 5))

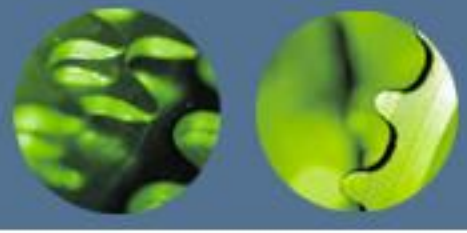
- 2 3 +
- 2 4 5 * +
- 2 4 + 7 * 3 9 5 - * +

- Infix vs postfix
- Why do we want to do this transformation?



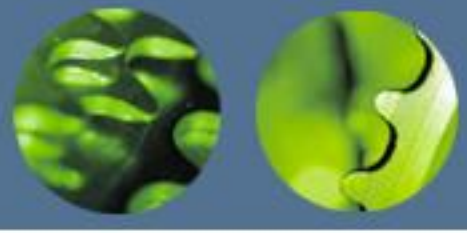
Infix to postfix

- Read ch from input until empty
  - If ch is arg , output = output + arg
  - If ch is “(“, push ‘(‘;
  - If ch is op and higher than top push ch
  - If ch is “)” or end of input,
    - output = output + pop() until empty or top is “(“
  - Read next input



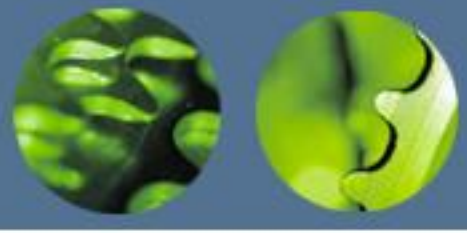
Postfix eval

- 5 + 2 * 3 -> 5 2 3 * +
- Algorithm
  - While input is not empty
  - If ch is number , push (ch)
  - Else
    - Pop (a)
    - Pop(b)
    - Eval (ch, a, b)



Quick XML Review

- XML – Wave of the future



Another Real world example

- <?xml version = "1.0"?>
- <!-- An author -->
- <author>
- <name gender = "male">
- <first> Art </first>
- <last> Gittleman </last>
- </name>
- </author>