Queues

- DEMO LESSON
- DR. MICHAEL ROBBELOTH
- TARGET COURSE: DATA STRUCTURES
- APRIL 8, 2025

ACM CS 2023 Coverage

- Algorithms and Complexity (AL)
  - Fundamental Data Structures and Algorithms (CS Core: 24 hours, KA Core: 7 hours overall, one hour on queues) pg 18, 31-32
- SDF/Fundamental Data Structures (12 core Tier-1 hours), pg. 305
- Outcome: Know when to select a queue as an appropriate data structure for a given problem, pgs. 307-308
- Outcome: Write programs that use a key abstract data type such as a queue, pgs 307-308

ABET Coverage

- ABET CAC Criterion 2 & 6
    - Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
    - Apply computer science theory and software development fundamentals to produce computing-based solutions.
- ABET EAC Criterion 1 & 7
    - an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
    - an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

Today

- Introduction to Queues
- Queue Characteristics
- Queue Operations
- Types of Queues
- Queue Implementation
- Applications of Queues and Practical Examples
- Quiz

Lines

- “An Englishman, even if he is alone, forms an orderly queue of one.”
- George Mikes (British Writer)

Where have you encountered a wait?

Hurry up and Wait.

- Waiting to ride a roller coaster
- Waiting at the ATM to make a withdrawal
- Waiting in the security line at an airport*
- Processing through immigration*

Hurry up and Wait.

- YOU'VE BEEN IN A REAL-LIFE IMPLEMENTATION OF A QUEUE.
- TODAY, WE'LL FOCUS ON THE VIRTUAL ONES

So, what is a queue?

- A collection of items that are placed into a structure *in some order* at one end
- Removed from the structure at the other end
- The entry point is called the tail/rear/back
- The exit point is called the head/front
- In theory, no storage limits (of course, applied computer scientists have to deal with resource limits)

Today

- Introduction to Queues
- Queue Characteristics
- Queue Operations
- Types of Queues
- Queue Implementation
- Applications of Queues and Practical Examples
- Quiz

Queue Characteristics 1/2

- A type of sequence (elements in a linear order)
- First-in First-out (FIFO) data structure
- The item at the front/rear of the queue has been there the longest/shortest
- Search (Average/Worst) O(n)/O(n)
- Insert/Delete O(1)/O(1) (if linked-list implementation; array is O(n))
- Access O(n) (if linked-list implementation; array is O(1))
- Space O(n)/O(n)

Queue Characteristics 2/2

- For Java, it doesn't handle null objects, indicating an empty queue (some implementations may not allow it)
- Adding to a full/empty queue is overflow/underflow
- Can be used in place of a tape in a Turing machine with two heads moving to the right
  - Semi-infinite work tape
  - The leading head is write-only
  - The right head is read only
  - Thus, FIFO storage
  - Can simulate a normal Turing machine (not true of pushdown stores or counters)

“How is a queue different from a stack?

- FIFO vs LIFO
- Stacks are better for undo operations, calculating expressions, and backtracking in algorithms
- Interact at rear and front vs just at the front (top)

Today

- Introduction to Queues
- Queue Characteristics
- Queue Operations
- Types of Queues
- Queue Implementation
- Applications of Queues and Practical Examples
- Quiz

Enqueue

ADDING AN ELEMENT
TO THE REAR

Dequeue

REMOVING AN ELEMENT FROM THE FRONT

Peek (Front)

LOOK AT THE FRONT OF THE QUEUE WITHOUT REMOVING THE ELEMENT

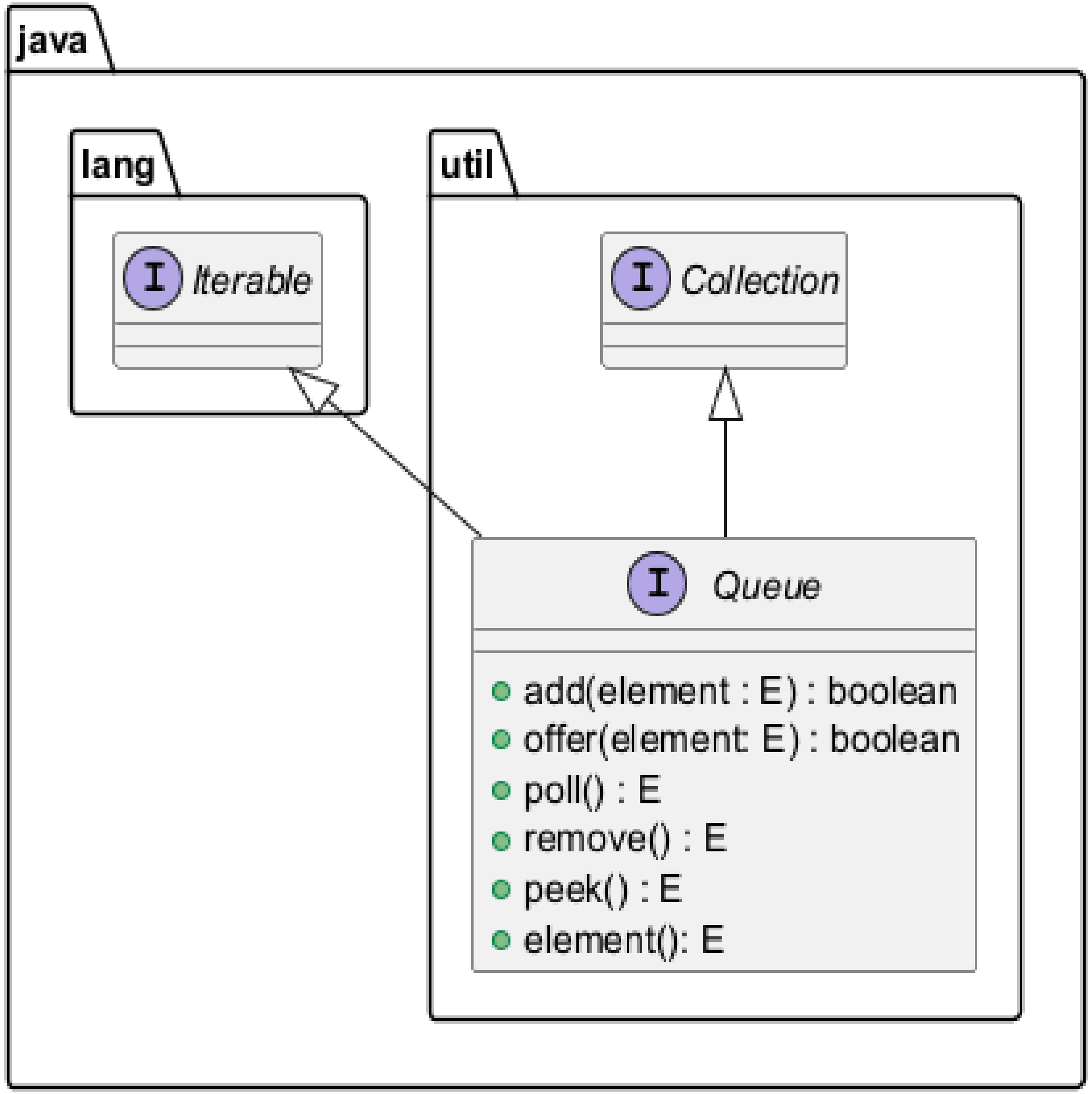
Empty/isEmpty

CHECK TO SEE IF QUEUE CONTAINS ANY ELEMENTS

Full/isFull

UNABLE TO ACCEPT NEW ELEMENTS FOR ENQUEUEING

Java Queue Interface



Today

- Introduction to Queues
- Queue Characteristics
- Queue Operations
- Types of Queues
- Queue Implementation
- Applications of Queues and Practical Examples
- Quiz

Types of Queues

- Linear Queue (Basic)
- Circular Queue (Try to limit wasted space)
- Priority Queue (Elements are assigned a priority)
  - Operations: INSERT, MAX, REMOVE, REMOVEMAX, MODIFY, and SIZE
  - INSERT/MODIFY O(N) given need to find the correct spot in queue for insertion/change (if sorted)
  - MAX/REMOVEMAX O(N) if unsorted
- Double Queue (Deque in Java, insert/remove at both ends)

Circular Queue Example with Array

0 1 2 3 4 5 6 7

- Queue: 51, 81, 21, 87, 13, 46, 28, **blank**; Front = 0, Rear = 7, count = 7
- After dequeue of 51 (51 is still in queue, but can be overwritten now, just make a copy)
- Queue 51, 81, 21, 87, 13, 46, 28, blank; Front=1, Rear = 7, count = 6
- After dequeue of 81
- Queue 51, 81, 21, 87, 13, 46, 28, blank; Front=2, Rear = 7, count = 5
- Enqueue 43 (wrap around of rear)
- Queue 51, 81, 21, 87, 13, 46, 28, **43**; Front=2, Rear = 0 count = 6 (so index 0 is effectively blank)
- Enqueue 22, 91
- Queue **22, 91**, 21, 87, 13, 46, 28, 43; Front=2, Rear=2, count=8

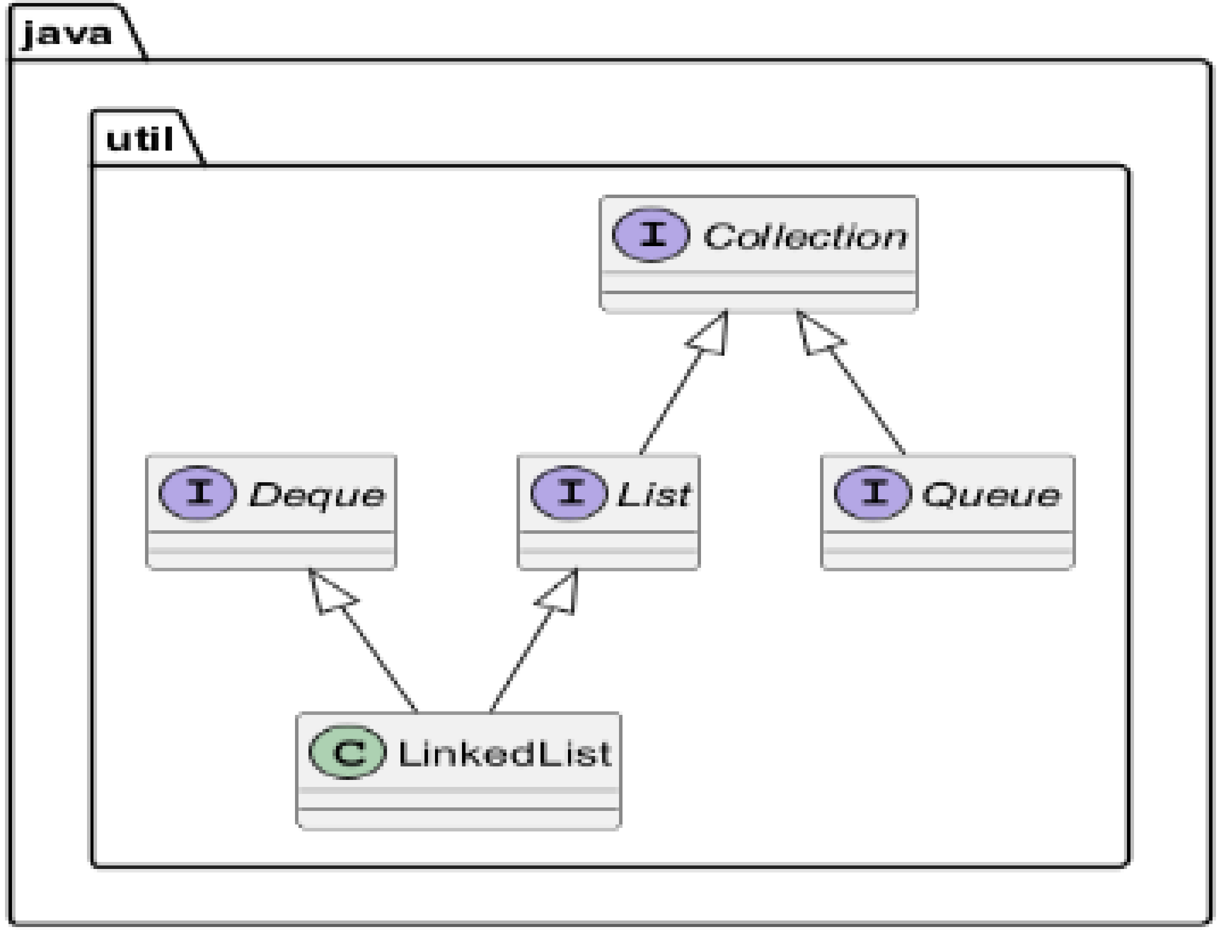
Today

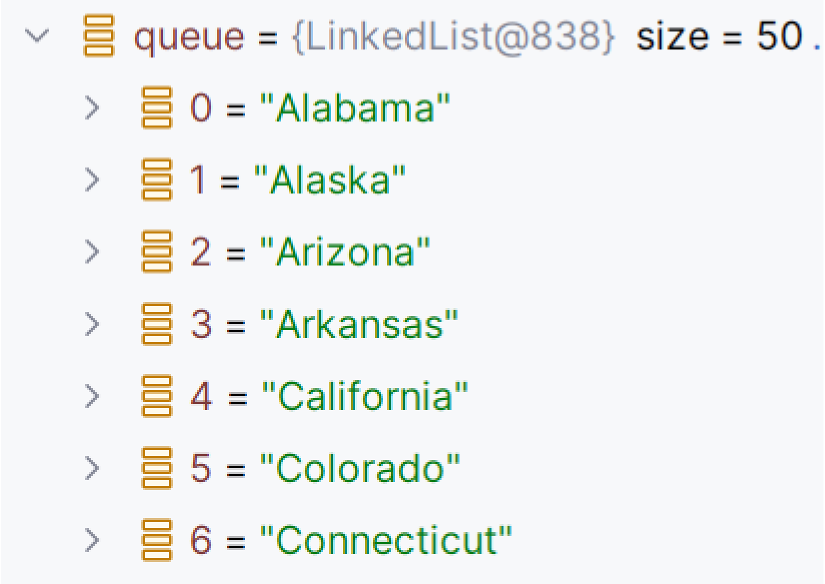
- Introduction to Queues
- Queue Characteristics
- Queue Operations
- Types of Queues
- Queue Implementation
- Applications of Queues and Practical Examples
- Quiz

Implementation

- Think: Buffer/Circular Buffer
- Lower Level: Array (fixed) or Singly/Doubly Linked List (dynamic)
- Alternative: Double Stack or Stack pointer & Base pointer

Java Implementation Classes – LinkedList Queue

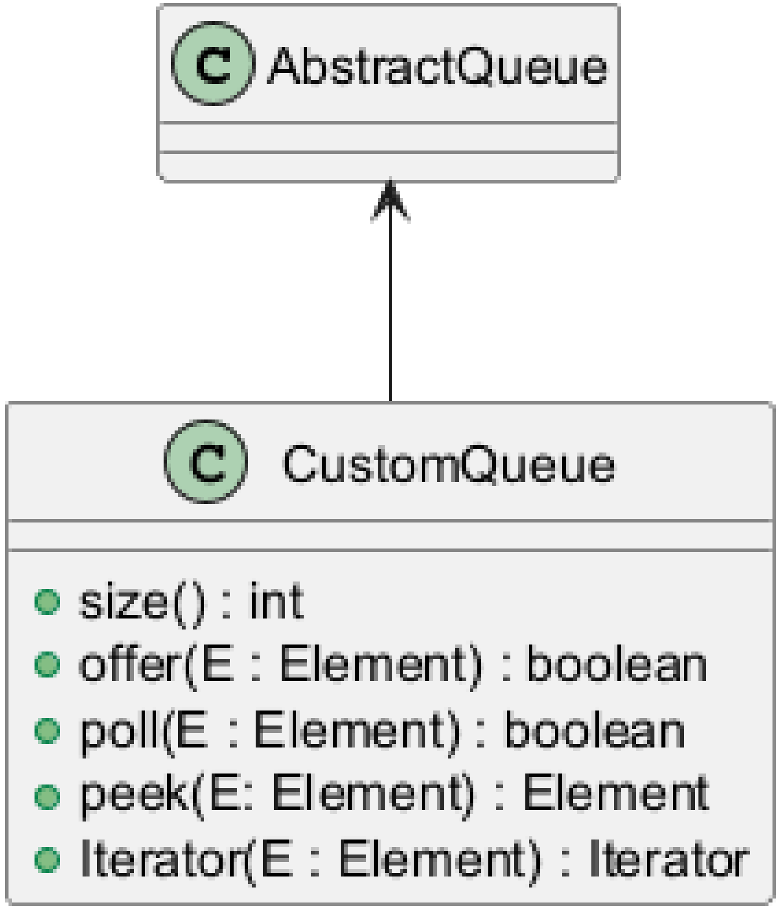




```
queue = {LinkedList@838}  size = 50 .
  0 = "Alabama"
  1 = "Alaska"
  2 = "Arizona"
  3 = "Arkansas"
  4 = "California"
  5 = "Colorado"
  6 = "Connecticut"
```

AbstractQueue

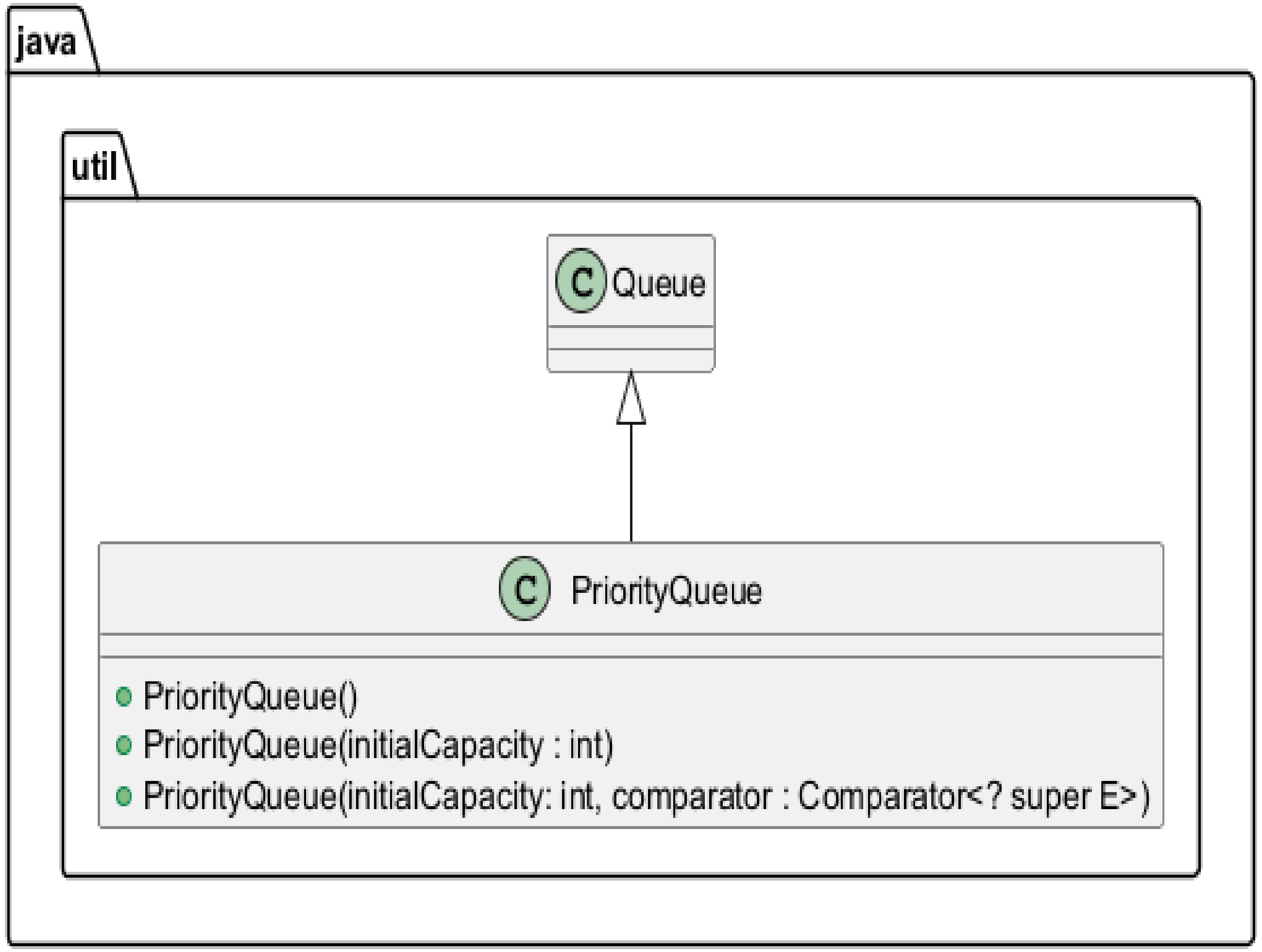
- Class for the implementation of a minimally viable queue class
- CustomQueue here overrides size, offer, pool, peek, and Iterator among other methods

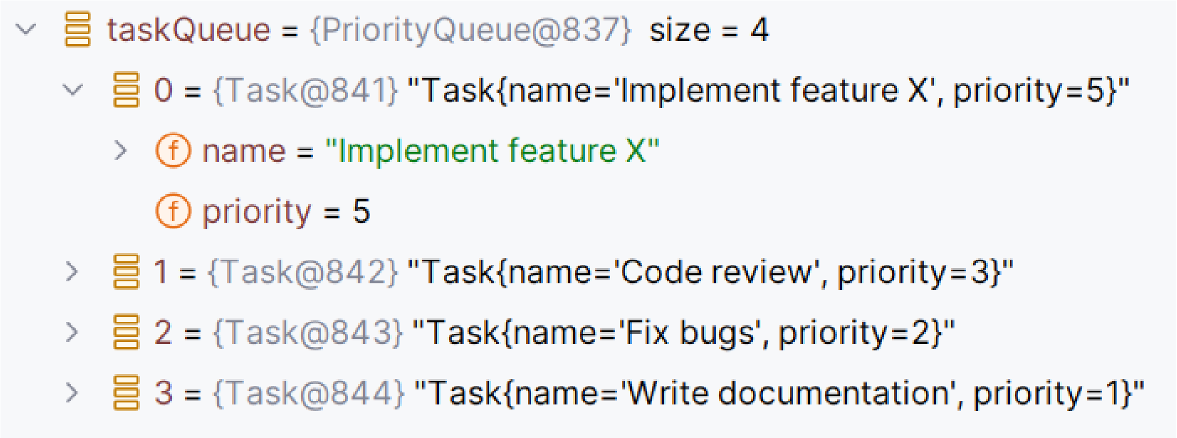

C AbstractQueue
C CustomQueue
size() : int
offer(E : Element) : boolean
poll(E : Element) : boolean
peek(E: Element) : Element
Iterator(E : Element) : Iterator

Blocking Queue

- Can't use the regular queue in concurrent operations w/o providing some synchronization mechanism to access them
- An alternative is to use a Blocking Queue
- Array, LinkedList, Priority variants
- Let's see an example using ArrayBlockingQueue

Java Implementation Classes – Priority Queue


java
util
C Queue
C PriorityQueue
PriorityQueue()
PriorityQueue(initialCapacity : int)
PriorityQueue(initialCapacity: int, comparator : Comparator<? super E>)



```
taskQueue = {PriorityQueue@837}  size = 4
  0 = {Task@841} "Task{name='Implement feature X', priority=5}"
    name = "Implement feature X"
    priority = 5
  1 = {Task@842} "Task{name='Code review', priority=3}"
  2 = {Task@843} "Task{name='Fix bugs', priority=2}"
  3 = {Task@844} "Task{name='Write documentation', priority=1}"
```

Other Interesting Java Queues

- DelayQueue – task scheduling, expiry caching, and rate limiting
- SynchronousQueue – can't enqueue an item until another item is dequeued on another thread
- There are others…

Today

- Introduction to Queues
- Queue Characteristics
- Queue Operations
- Types of Queues
- Queue Implementation
- Applications of Queues and Practical Examples
- Quiz

Scheduling

OPERATING SYSTEMS

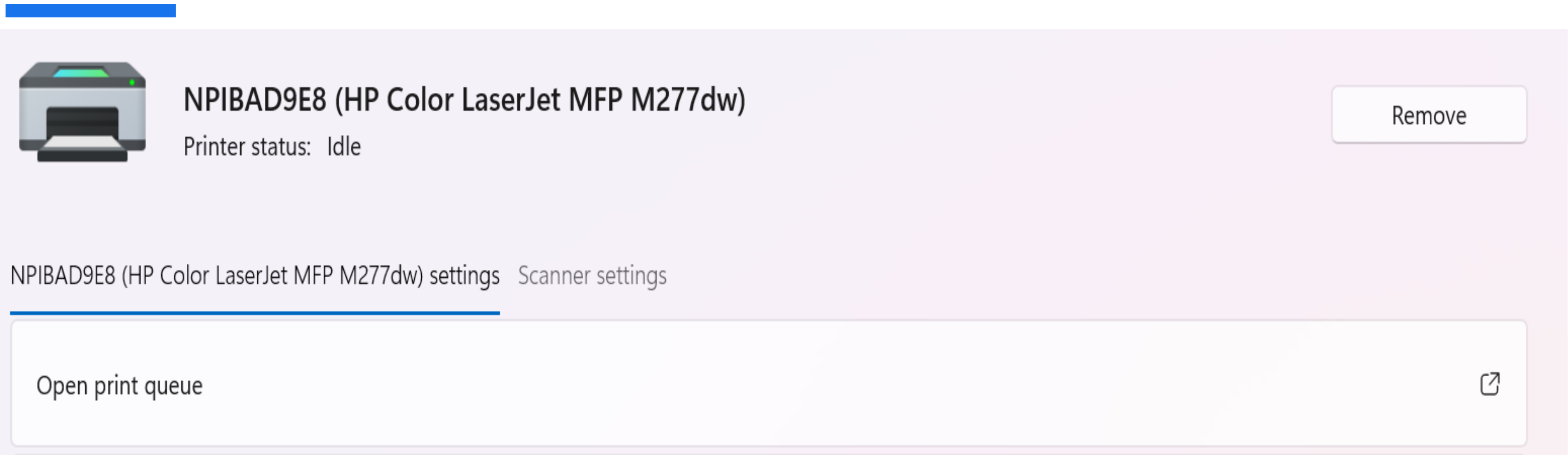
READY QUEUES – USE FIFO, SHORTEST JOB NEXT, OR ROUND ROBIN

JOB QUEUES – PROCESSES AWAITING EXECUTION

DEVICE QUEUES – PROCESSES NEEDING RESULTS OF I/O OPERATIONS

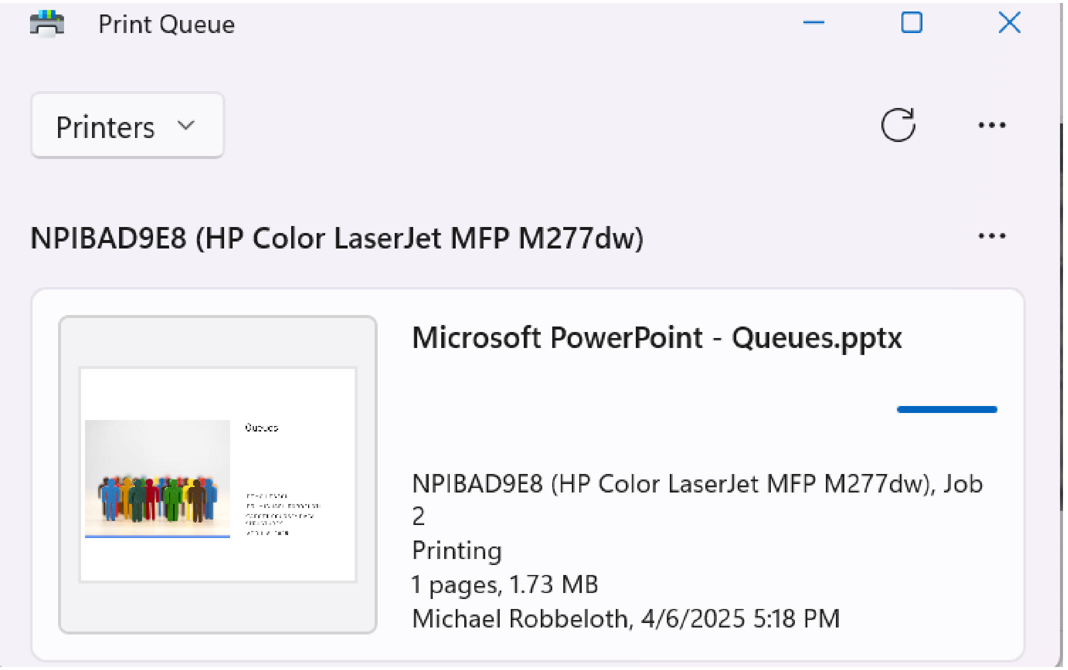
Print Job Management

- PRINT JOBS GO TO SPOOL
- FIFO ORDERING
- JOBS HELD UNTIL DEVICE IS READY
- PRINTS JOBS ONE AT A TIME



NPIBAD9E8 (HP Color LaserJet MFP M277dw)

Printer status: Idle

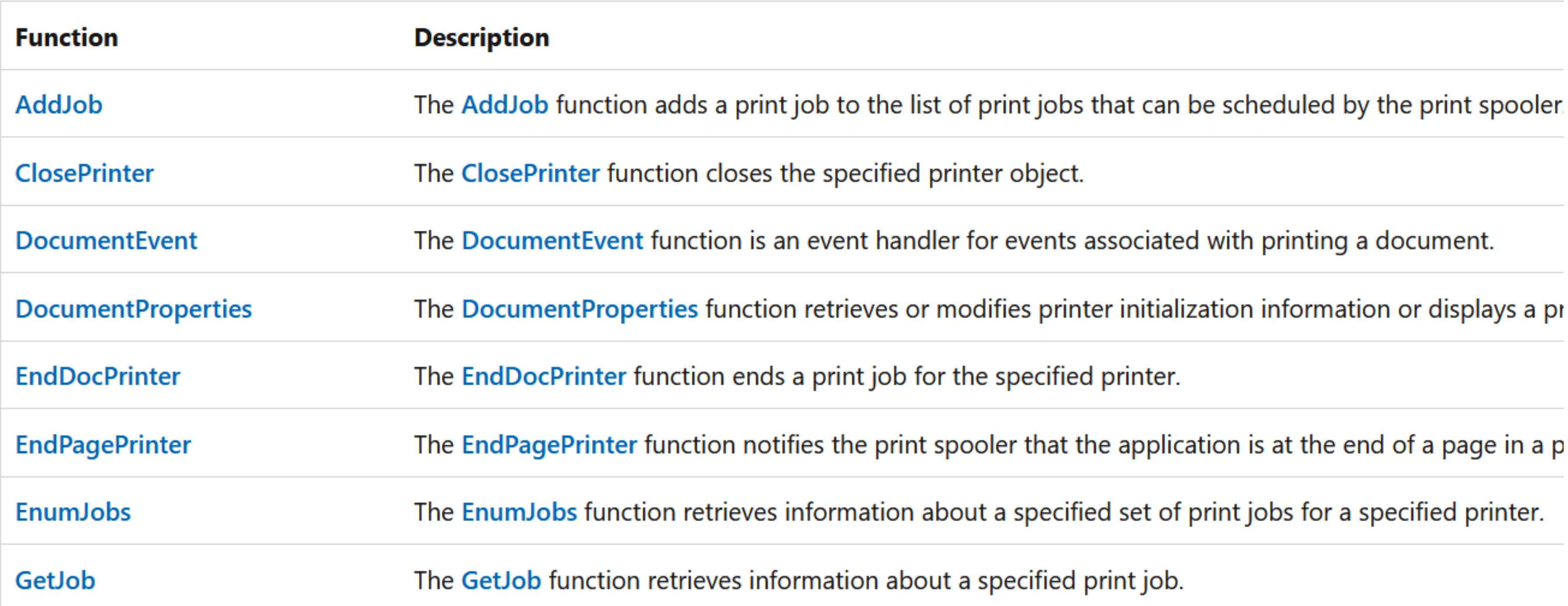


NPIBAD9E8 (HP Color LaserJet MFP M277dw)

Microsoft PowerPoint - Queues.pptx

NPIBAD9E8 (HP Color LaserJet MFP M277dw), Job 2
Printing
1 pages, 1.73 MB
Michael Robbeloth, 4/6/2025 5:18 PM

Win32 Print Spooler API Partial List



Function	Description
AddJob	The AddJob function adds a print job to the list of print jobs that can be scheduled by the print spooler
ClosePrinter	The ClosePrinter function closes the specified printer object.
DocumentEvent	The DocumentEvent function is an event handler for events associated with printing a document.
DocumentProperties	The DocumentProperties function retrieves or modifies printer initialization information or displays a pr
EndDocPrinter	The EndDocPrinter function ends a print job for the specified printer.
EndPagePrinter	The EndPagePrinter function notifies the print spooler that the application is at the end of a page in a p
EnumJobs	The EnumJobs function retrieves information about a specified set of print jobs for a specified printer.
GetJob	The GetJob function retrieves information about a specified print job.

Breadth-First Search (BFS)

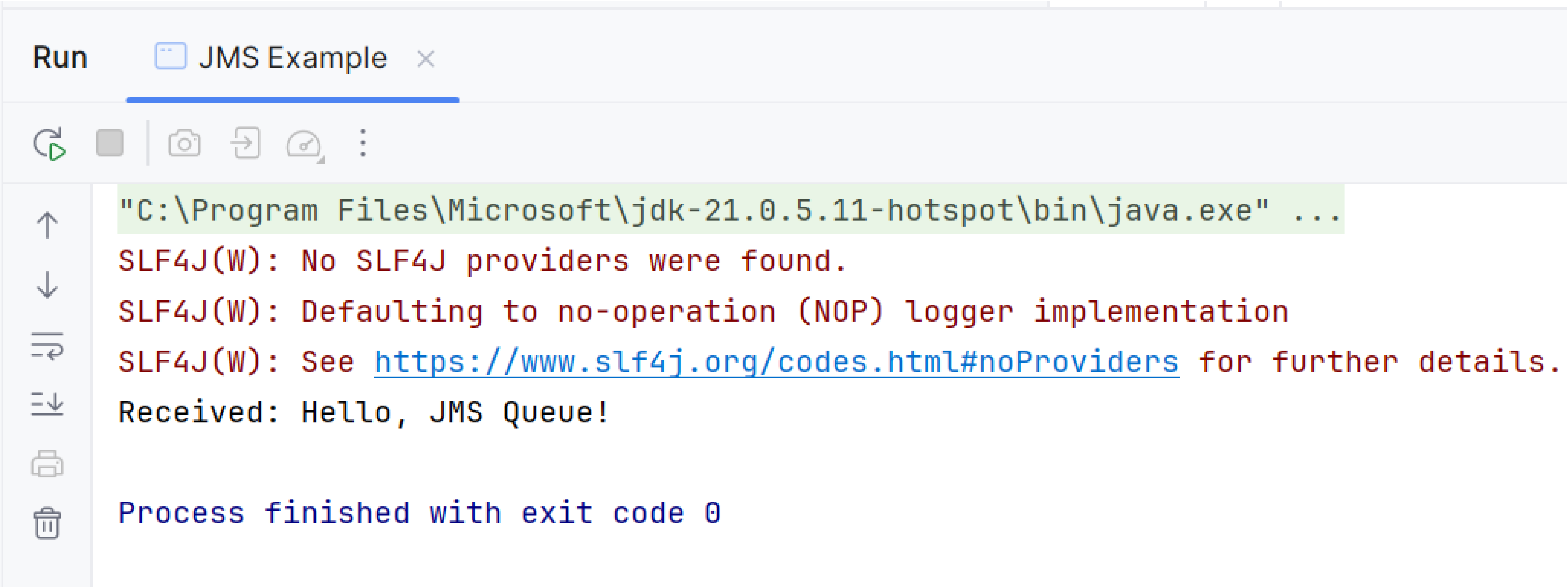
GRAPH ALGORITHMS

- VISITS NODES LEVEL-BY-LEVEL
- VISIT NODE, ADDS UNVISITED NEIGHBORS TO THE QUEUE
- VISITS UNVISITED NEIGHBORS IN FIFO ORDER

Java Messaging Service (JMS)

- Used in enterprise messaging (middleware component).
- Create, send, receive, and read messages between endpoints
- Common set of interfaces and semantics
- Asynchronous communication between software components
- Loosen coupling between components
- Can use publish-subscribe (content hierarchy) or point-to-point (queues) messaging models
- Requires a provider like Apache's ActiveMQ (works with multiple programming languages)
- In turn, non-Java/JMS systems can use proprietary APIs like CORBA from OMG to talk to JMS

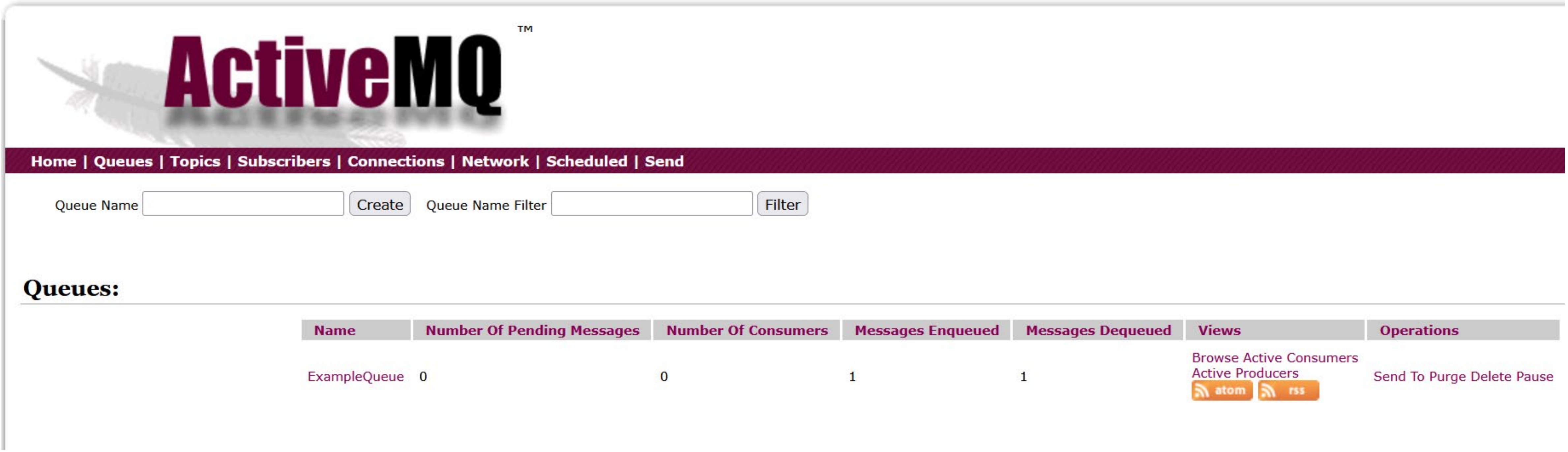
Java Messaging Service (JMS)



```
"C:\Program Files\Microsoft\jdk-21.0.5.11-hotspot\bin\java.exe" ...
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Received: Hello, JMS Queue!

Process finished with exit code 0
```

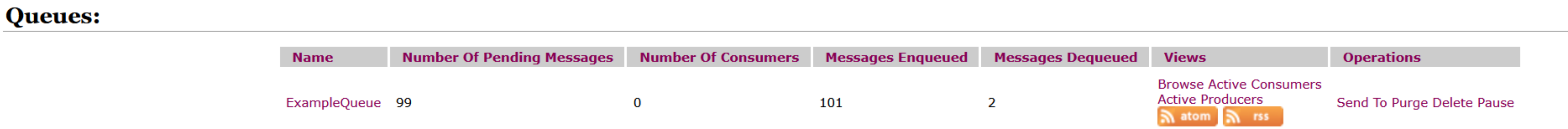
Java Messaging Service (JMS)


ActiveMQ™
Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send
Queue Name
Create
Queue Name Filter
Filter

Queues:

Name	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
ExampleQueue	0	0	1	1	Browse Active Consumers Active Producers atom rss	Send To Purge Delete Pause

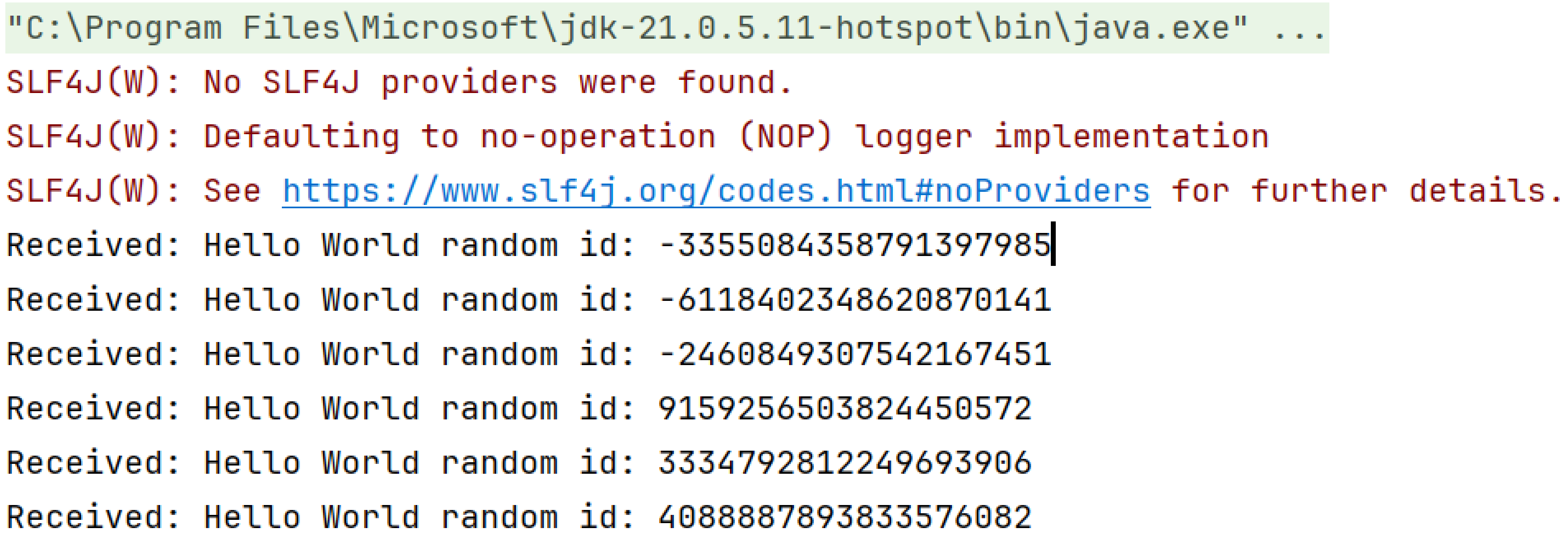
Java Messaging Service (JMS)



Queues:

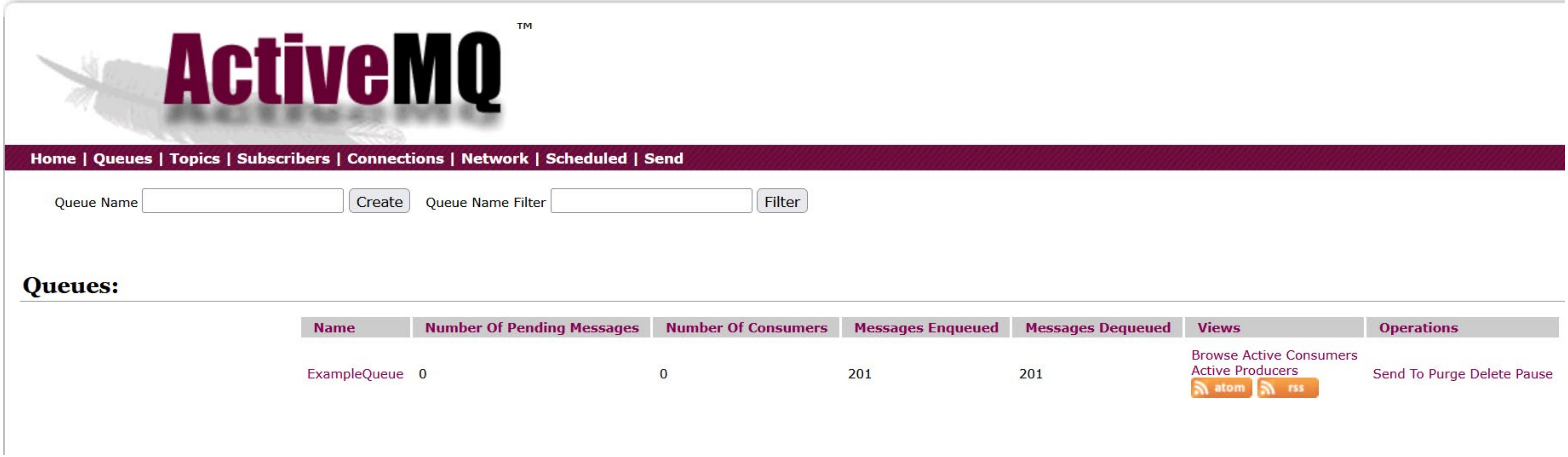
Name	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
ExampleQueue	99	0	101	2	Browse Active Consumers Active Producers atom rss	Send To Purge Delete Pause

Java Messaging Service (JMS)



```
"C:\Program Files\Microsoft\jdk-21.0.5.11-hotspot\bin\java.exe" ...
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Received: Hello World random id: -3355084358791397985
Received: Hello World random id: -6118402348620870141
Received: Hello World random id: -2460849307542167451
Received: Hello World random id: 9159256503824450572
Received: Hello World random id: 3334792812249693906
Received: Hello World random id: 4088887893833576082
```

Java Messaging Service (JMS)


ActiveMQ™
Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send
Queue Name
Create
Queue Name Filter
Filter

Queues:

Name	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
ExampleQueue	0	0	201	201	Browse Active Consumers Active Producers atom rss	Send To Purge Delete Pause

Today

- Introduction to Queues
- Queue Characteristics
- Queue Operations
- Types of Queues
- Queue Implementation
- Applications of Queues and Practical Examples
- Quiz

Quiz is in a Kahoot…give me a minute