**WIA1002 TUTORIAL 7**

**QUEUE**

1. Name **four** everyday examples of a queue other than those discussed during lecture.

Answer: Cars waiting in a line at a toll booth or a gas station

Message waiting to be sent in an email server’s outgoing message queue.

Customers waiting in a queue for customer service at a call center

Package waiting to be shipped in a postal service’s outgoing package queue.

2. What is the difference between a queue and stack?

Answer:

For ordering part, queue use FIFO but stack use LIFO.

For operation part, queue support two main operation that is enqueue and dequeue but stack support push and pop.

For usage, queue typically used in scenarios where the order of processing or service matter but stacks used in scenarios where the last item added is the first item to be processed.

3. Use the following code segment to answer parts (a) through (c):

Queue<Integer> q = new Queue<Integer>();

Scanner keyIn = new Scanner(System.in);

for (int i = 1; i <= 5; i++)

{

if (keyIn.nextBoolean())

System.out.print(i + " ");

else

q.enqueue(i);

}

}

while (!q.isEmpty())

System.out.print(q.dequeue() + " ");

System.out.println();

1. (a) What is the output for the following input sequence?

true false false true true

1. (b) Is it possible to have output: 1 3 5 4 2? If yes, give an input sequence that produces the output; or else, provide justification to your answer.

(c) Give at least **three** input sequences that produce the output: 1 2 3 4 5

Answer:

1. Output: 1 4 5 2 3
2. No, since FIFO, so 2 must be print before 4 if use true false true false true
3. True true true true true

False false false false false

True true true true false

4. Hand trace a queue X through the following operations:

X.enqueue(new Integer(14));

X.enqueue(new Integer(3));

X.enqueue(new Integer(5));

Object Y = X.dequeue();

X.enqueue(new Integer(7));

X.enqueue(new Integer(9));

Y = X.dequeue();

X.enqueue(new Integer(2));

X.enqueue(new Integer(4));

Given the resulting queue X above, what would be the result of each of the following?

1. a) X.front();
2. b) Y = X.dequeue();

X.enqueue(new Integer(10));

X.front();

1. c) Y = X.dequeue();
2. d) X.front();

Answer:

Y: 14 3 5 7 (wrong? Y not a queue, will be totally replacing by the X.dequeue)

X= 9 2 4 10

1. 5 (Y take 2element from X)
2. 7 (Y take one again)
3. Y=7
4. 9

5. Provide and explain three operations / functions of Linked List based Queue that you can add to the GenericQueue class, other than those discussed in the lecture (i.e., enqueue, dequeue and getSize – from Slide 10 in the lecture slide).

The first operation that can be added is the peek() method, where it returns the first element in the queue without actually removing it.

The second operation that can be added is the isEmpty() method, where it returns a boolean value to check if the queue is empty.

The third operation that can be added is the clear() method, where it clears the queue by setting the head and the tail pointer towards null.

**WIA1002 TUTORIAL 7b**

**PRIORITY QUEUE**

1. Describe the main difference between Queue and PriorityQueue.

2. Briefly provide THREE (3) real-life example in using PriorityQueue.

3. Show the output for every System.out.println ((a) – (f)) in the following code:

|  |
| --- |
| import java.util.\*;  public static void main(String args[])  {  PriorityQueue<String> pQueue = new PriorityQueue<String>();  pQueue.offer("C++");  pQueue.offer("Python");  pQueue.offer("Java");  pQueue.offer("Fortran");  System.out.println("peek() gives us: "+ pQueue.peek()); //(a)  System.out.println("The queue elements:"); //(b)  Iterator itr = pQueue.iterator();  while (itr.hasNext())  System.out.println(itr.next()); //(b)  pQueue.poll();  System.out.println("After poll():"); //(c)  Iterator<String> itr2 = pQueue.iterator();  while (itr2.hasNext())  System.out.println(itr2.next()); //(c)  pQueue.remove("Java");  System.out.println("After remove():"); //(d)  Iterator<String> itr3 = pQueue.iterator();  while (itr3.hasNext())  System.out.println(itr3.next()); //(d)  boolean b = pQueue.contains("Ruby");  System.out.println ( "Priority queue contains Ruby or not?: " + b); //(e)  Object[] arr = pQueue.toArray();  System.out.println ( "Value in array: "); //(f)  for (int i = 0; i<arr.length; i++)  System.out.println ( "Value: " + arr[i].toString()) ; //(f)  } |

ANSWER:

1. In queue use FIFO, but in priority queue does not strictly follow FIFO, elements are assigned with priority, element with highest priority is removed first regardless of the order.
2. Emergency on road will let car with priority go first

In air traffic control system, planes are given priority based on their level of emergency, distance to destination and available resource.

Search algorithms, such as Dijkstra's algorithm, use priority queues to determine the shortest path between two points. The priority queue is used to store the next vertex with the smallest distance, allowing the algorithm to traverse the graph in an efficient manner.

To-do-list, the tasks with higher priority or deadlines can be listed at the top to ensure they are completed first.

1. a) peek() gives us: C++

b) The queue elements:

C++

Fortran

Java

Python

1. After poll():

Fortran

Java

Python

1. After remove():

Fortran

Python

1. Priority queue contains Ruby or not?: False
2. Value in array:

Value: Python

Value: Fortran

4. Answer the following sub-questions with referring to the following code:

|  |
| --- |
| public class PriorityQueue2 {  public static void main(String... args ){  PriorityQueueComparator pqc=new PriorityQueueComparator();  PriorityQueue<String> pq=new PriorityQueue<String>(5,pqc);  pq.add("Jason");  pq.add("Ali");  pq.add("Muhamad");  for(String s:pq){  System.out.println(s);  }  }  }  public class PriorityQueueComparator implements Comparator<String>{  public int compare(String s1, String s2) {  if (s1.length() < s2.length()) {  return -1;  }  if (s1.length() > s2.length()) {  return 1;  }  return 0;  }  } |

1. a) What is the purpose of the PriorityQueueComparator in the code?
2. b) What is the output for the code?

ANSWER:

1. to compare the length of string and return integer to arrange the input according to their length. Purpose of PriorityQueueComparator is to define a custom ordering method for the elements of the PriorityQueue. (shorter length get priority)
2. Ali

Jason

Muhamad