**Queue->**

Explantion->

Child interface of collection.

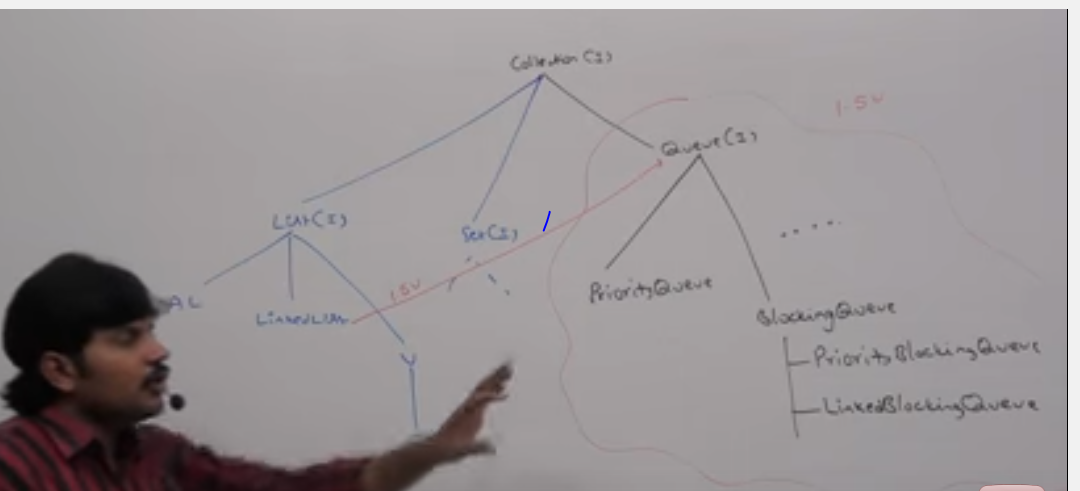
First in first out..

Queue

Priority queue blocking queue

Priority blocking queue

Linked blocking queue



1.5 version enhancements {queue intraface}.

It’s the child interaface of collection.

If we want to represent a group of individual objects prior to processing then we shuld go for queue.

Ex->before sending sms message all mobile numbers in some data structure.in which order we added mobile numbers in the same order only message should be send. For this “fifo” requirement queue is a best choice.

Usally queue follows fifo.but based on our requirement we can implement our own priority order also{:”priority queue”}.

From 1.5 ver onward linked list class also implements the queue intraface.

That use fifo.

Queue Method are five specific methods->

1.offer(object a);//for adding an element.

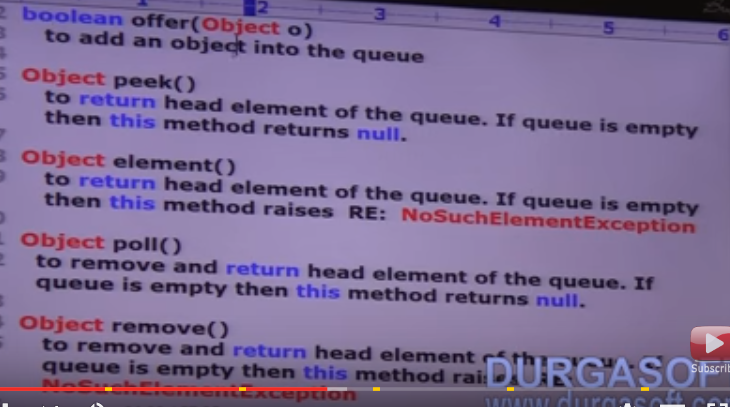
2. pull();//for remove an element if queue is empty the nit simply return null.

3.remove();///removie the heads and if there is nohead to remove then it give a exeption.

4.peek();//for get head element

if queue is empty then it simply returns null.

5.element();//if queue is empty it gives error run time exeption



Priority queue->

If we want to represent a group of individual objects.prior to processing.

According to some priority.then we should go for priority queue.the priority can be

Either default natural sorting order or customize order define by compartor.

Intertion order is not preserved.and its based on some priority.

* Store elements according to some priority.
* Dnso//default sorting order.

C so //customize sorting order.

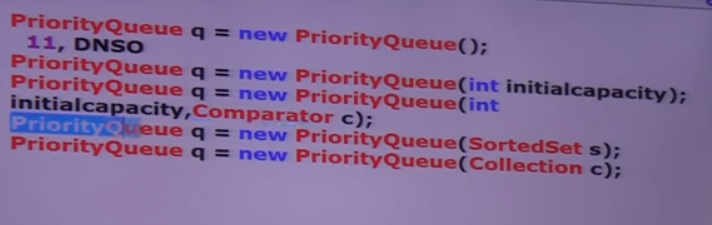
* Insertion order is not preserved.
* Dublicates are not allowed.

If we are depending on default natural sorting order compulsory the object should be homogenous and comparable other wise we will get run time exeption.

If we are defining our own sorting by comparator then object need not be homogenous and a comparable.

* null is not allowed even at the first element also.

Constructor



1.pq q=new pq();

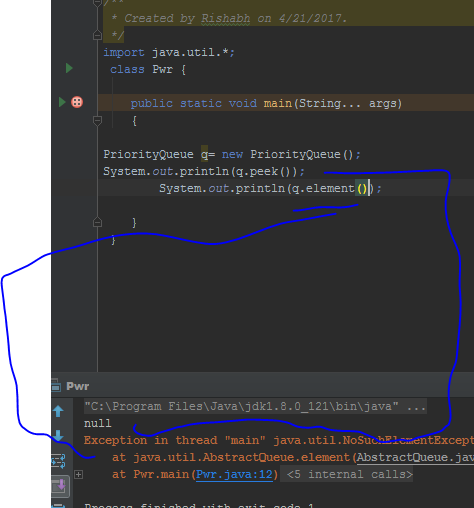
Creates a empty priority queue with intitial capacity11 and all object be inserted according to default natural sorting order.

2. pq q=new pq(int init);

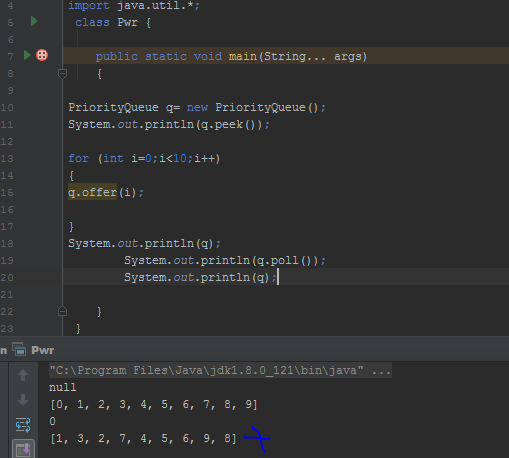
3. pq q=new pq(int init ,comparator c);

4.pq q= new pq (sortedset s);

5. pq q= new pq (collection c);



\*queue should not be empty..



Some platforms wont provide proper support for thread priorities.and priority queue.

So even in the original exam you shoul write1234….

