

Implementare una classe Queue di elementi di tipo Object che fornisca i metodi descritti di seguito utilizzando come contenitore un vettore (array) di oggetti di tipo Object.

Definire un costruttore con parametro ed un costruttore di default. Rispettare i requisiti anche in termini di eccezioni generate.

Method Summary

boolean	add (Object e) Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions, returning <code>true</code> upon success and throwing an <code>IllegalStateException</code> if no space is currently available.
Object	element () Retrieves, but does not remove, the head of this queue.
Boolean	offer (Object e) Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions.
Object	peek () Retrieves, but does not remove, the head of this queue, or returns <code>null</code> if this queue is empty.
Object	poll () Retrieves and removes the head of this queue, or returns <code>null</code> if this queue is empty.
Object	remove () Retrieves and removes the head of this queue.

Method Detail

add

boolean **add**(Object e)

Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions, returning `true` upon success and throwing an `IllegalStateException` if no space is currently available.

Parameters:

e - the element to add

Returns:

`true`

Throws:

[IllegalStateException](#) - if the element cannot be added at this time due to capacity restrictions

[ClassCastException](#) - if the class of the specified element prevents it from being added to this queue

[NullPointerException](#) - if the specified element is null and this queue does not permit null elements

[IllegalArgumentException](#) - if some property of this element prevents it from being added to this queue

offer

boolean **offer**([Object](#) e)

Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions. When using a capacity-restricted queue, this method is generally preferable to [add\(Object\)](#), which can fail to insert an element only by throwing an exception.

Parameters:

e - the element to add

Returns:

true if the element was added to this queue, else false

Throws:

[ClassCastException](#) - if the class of the specified element prevents it from being added to this queue

[NullPointerException](#) - if the specified element is null and this queue does not permit null elements

[IllegalArgumentException](#) - if some property of this element prevents it from being added to this queue

remove

[Object](#) **remove**()

Retrieves and removes the head of this queue. This method differs from [poll](#) only in that it throws an exception if this queue is empty.

Returns:

the head of this queue

Throws:

[NoSuchElementException](#) - if this queue is empty

poll

[Object](#) **poll**()

Retrieves and removes the head of this queue, or returns null if this queue is empty.

Returns:

the head of this queue, or null if this queue is empty

element

[Object](#) **element**()

Retrieves, but does not remove, the head of this queue. This method differs from [peek](#) only in that it throws an exception if this queue is empty.

Returns:

the head of this queue

Throws:

[NoSuchElementException](#) - if this queue is empty

peek

[Object](#) **peek()**

Retrieves, but does not remove, the head of this queue, or returns `null` if this queue is empty.

Returns:

the head of this queue, or `null` if this queue is empty