

# Implementación de Colas en Java

Estructuras de Datos  
Grado en Ingenieria Informatica  
Universidad de Burgos

Bruno Baruque Zanon y Pedro Renedo Fernandez

# Contenido

- 1 La clase *AbstractQueue*
- 2 Implementación del interfaz *Deque*

# La clase *AbstractQueue* (I)

- Implementación “esquelética” de la interfaz *Queue*.
- Las implementaciones proporcionadas son apropiadas siempre que no se permita la inserción de elementos nulos.
- Los métodos *add*, *remove* y *element* están basados en *offer*, *poll*, y *peek* respectivamente.

Diferencia: lanzan excepciones en lugar de devolver **false** o **null** al fallar

- En este caso SI es necesario implementar un iterador (No se proporciona ninguno).

A diferencia de en *AbstractList* y *AbstractSequentialList*

- *PriorityQueue* extiende esta clase.

# La clase *AbstractQueue* (II)

- Para implementar una cola NO modificable se necesitará sobrecribir los métodos *Collection.size()*, *Queue.peek()*, *Collection.iterator()*
- Puesto que la cola solo está definida para el acceso por un extremo, no resulta muy útil que existan colas no modificables.

A diferencia de las listas, que definen el acceso aleatorio.

- Para una lista modificable de tamaño variable, se necesita sobrecribir *Queue.offer(E)* (que no permita insertar nulos) y *Queue.poll()*.

# La clase *AbstractQueue* (III)

- Autores: Josh Bloch, Neal Gafter.

```
1 public abstract class AbstractQueue<E>
2     extends AbstractCollection<E> implements Queue<E> {
3
4     protected AbstractQueue() {
5     }
6
7     public boolean add(E e) {
8         if (offer(e))
9             return true;
10        else
11            throw new IllegalStateException("Queue_full");
12    }
```

# La clase *AbstractQueue* (IV)

```
1      public E remove() {  
2          E x = poll();  
3          if (x != null)  
4              return x;  
5          else  
6              throw new NoSuchElementException();  
7      }  
8  
9      public E element() {  
10         E x = peek();  
11         if (x != null)  
12             return x;  
13         else  
14             throw new NoSuchElementException();  
15     }
```

# La clase *AbstractQueue* (V)

```
1  public boolean addAll(Collection<? extends E> c) {  
2      if (c == null)  
3          throw new NullPointerException();  
4      if (c == this)  
5          throw new IllegalArgumentException();  
6      boolean modified = false;  
7      for (E e : c)  
8          if (add(e))  
9              modified = true;  
10     return modified;  
11 }
```

# La clase *AbstractQueue* (V)

```
1  public void clear() {  
2      while ( poll() != null)  
3          ;  
4  }
```



# Implementación del interfaz *Deque*

- En este caso, Java NO proporciona ninguna clase abstracta para su implementación.

Por ejemplo, la clase *ArrayDeque* extiende solo la clase *AbstractCollection* (e implementa *Deque*)

# Referencias (I)

## En internet

[Oracle, 2015c] Tutorial de las colecciones de Java.

[Oracle, 2015a] Documentación de las colecciones de Java.

[Oracle, 2015b] Preguntas y respuestas frecuentes sobre las colecciones de Java.

[Oracle, 2010] Ejemplos de uso de las colecciones.

# Referencias (II)

## Libros

[Naftalin and Wadler, 2007] Libro sobre la genericidad en Java y sus colecciones. El capítulo 15 está dedicado a las listas.

[Collins, 2011] En el capítulo 6 se describe *ArrayList* y en el 7 *LinkedList*.

# Referencias (III)



Collins, W. J. (2011).  
*Data Structures and the Java Collections Framework*.  
Wiley, 3 edition.  
<http://www.wiley.com/college/collins>.



Naftalin, M. and Wadler, P. (2007).  
*Java Generics and Collections*.  
O'Reilly.



Oracle (2010).  
Collections framework code samples.  
Sun Developer Network.  
<http://java.sun.com/developer/codesamples/collections.html>.



Oracle (2015a).  
Annotated outline of collections framework.  
<http://docs.oracle.com/javase/8/docs/technotes/guides/collections/reference.html>.

# Referencias (IV)



Oracle (2015b).

Java collections API design FAQ.

[http:](http://docs.oracle.com/javase/8/docs/technotes/guides/collections/designfaq.html)

[//docs.oracle.com/javase/8/docs/technotes/guides/collections/designfaq.html](http://docs.oracle.com/javase/8/docs/technotes/guides/collections/designfaq.html).



Oracle (2015c).

The java tutorials, trail: Collections.

<http://docs.oracle.com/javase/tutorial/collections/>.

# Licencia



Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional