



PRÁCTICA 02

La práctica corresponde al

Tema: Estructuras de Datos Lineales

Peso Total en la nota:
6%

Objetivos

Con esta práctica se pretende que el alumno se familiarice y comprenda:

1. Estructura de la jerarquía adicional del *Java Collections Framework*
 - a. Se estudia la forma de extender el Framework por medio de interfaces adicionales (Queue, Deque).
2. Cómo se implementa una estructura de datos relativamente sencilla de forma completa.
3. El coste computacional que implica realizar una estructura de datos con una estructura interna determinada.

Adicionalmente se trabajará con la Genericidad de Java, que permite definir los tipos de datos a almacenar en una colección. Es una característica que mantienen todos los componentes del *Collections Framework*.

Problema planteado

Se pretende generar un tipo de Cola nuevo para que quede a disposición de otros programadores dentro del *Java Collections Framework*.

El objetivo es programar una **Estructura de Datos Lineal** que combine las que se tienen por defecto en el *Java Collections Framework*. Adicionalmente se desea que esta estructura se modele como una Cola, para que el alumno se familiarice también con el funcionamiento de éstas.

Para esta implementación se pide la creación de una clase de nombre ColaMixta, que deberá extender la clase `java.util.AbstractQueue`. Se incluirá en el paquete `es.ubu.inf.edat.pr02_1920`. Su estructura interna debe ser la que aparece en el **Tema 3. Estructuras lineales: listas, pilas y colas**, como "Implementación Mixta" (transp. 26). Esto implica que se tratará de una estructura enlazada en la que, en cada nodo o segmento, se podrán almacenar varios elementos.

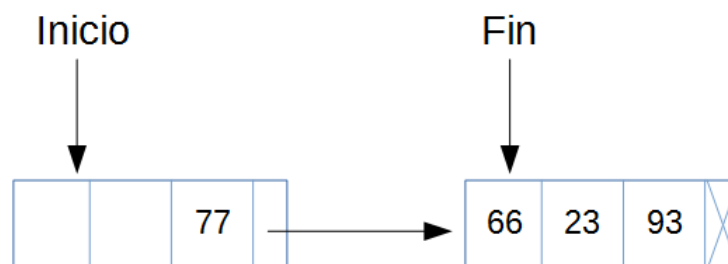


Ilustración 1: Estructura lineal mixta



Es interesante que la clase guarde una referencia al primer nodo (para extraer elementos) y al último (para insertarlos).

La clase ColaMixta deberá incluir una clase interna de nombre `NodoMixto<E>`, que permitirá construir cada uno de los nodos que componen la lista. Esta clase deberá contener los datos del nodo, así como las referencias que permitan enlazar a los demás nodos para construir la cola completa.

NodoMixto<E>

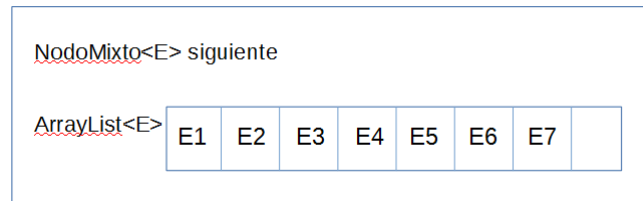


Ilustración 2: Estructura interna de NodoMixto

Por el hecho de implementar la clase `AbstractQueue`, también se pedirá incluir una clase interna para implementar el interfaz `Iterator` que permita recorrer los elementos almacenados en la cola.

Como Indicaciones adicionales a la práctica se incluyen:

1. El constructor de la clase `ColaMixta`, deberá recibir como parámetro el número de elementos que se pueden almacenar en cada nodo (serán todos del mismo tamaño). Este tamaño permanecerá invariable a lo largo de todo el programa.
2. Además de los métodos que se piden implementar por `AbstractQueue` (`offer`, `peek`, `poll`, `size` e `iterator`), se solicita un método de consulta adicional:

`E peek (int index)`

Este método permite consultar el elemento que se obtendría en la cola al realizar un número de operaciones de `poll` seguidas, que viene marcado por el parámetro `index`.

Ejemplo: en la `ColaMixta` de contenido `[a, b, c, d, e]` la llamada a `peek(2)`, devolverá el elemento `'c'`.

Se sugiere pensar bien la implementación del método para que aproveche al máximo la estructura interna de la **cola mixta**, evitando operaciones innecesarias.

Criterios de Evaluación

Dada la naturaleza y objetivo de la práctica, se tendrá en cuenta para su valoración (de más importante a menos):

1. La corrección del funcionamiento. Se valorará como aspecto más importante que el funcionamiento de la estructura de datos implementada cumpla con todas las condiciones solicitadas, tanto en la documentación oficial como en el enunciado.
2. La complejidad algorítmica de la solución aportada. Se deberá tener cuidado de implementar la respuesta por medio de soluciones que impliquen la menor complejidad siempre que esto sea posible.
3. La corrección del código. Ausencia de problemas como *warnings*, variables no empleadas, código inalcanzable, etc.
4. La documentación (formato Javadoc) y comentarios que incluya el alumno en el código



entregado.

Condiciones de Entrega

La práctica se realizará en grupos de 1 o 2 personas.

La entrega se realizará SOLAMENTE por medio de la plataforma UBUVirtual. Cada miembro del grupo deberá subir su propia copia de la solución. No se admitirán soluciones fuera de plazo.

La entrega incluirá la información de los miembros de la pareja de prácticas, tanto en los ficheros de código fuente como en los documentos asociados.

Cada entrega consistirá en un fichero comprimido (formato .zip), con la estructura de nombre: Apellidos1Nombre1_Apellidos2Nombre2.zip

Deberá que incluir dentro:

- Código fuente de la solución (dentro de la estructura de paquetes necesaria)
- Comentarios dentro del código: El fichero fuente deberá estar debidamente comentado con todas las aclaraciones necesarias para la mejor comprensión de la solución.
- Breve informe que justifique, de forma razonada, cuál se considera la complejidad algorítmica de cada método implementado por el alumno.

No es necesaria la entrega de ficheros binarios.