

Estructuras de datos

Clase práctica 6



Contenido

- Colas y pilas

Material elaborado por: Julián Moreno

Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión

Implementación mediante listas

Una forma para trabajar con colas y pilas es tratarlas igual que una lista enlazada sólo que enfocándose en los siguientes métodos:

```
LinkedList<Integer> C = new LinkedList<Integer>();  
LinkedList<Integer> P = new LinkedList<Integer>();
```

También puede ser Byte, Short, Long, Float, Double

	Cola	Pila
push	add(e)	add(e)
peek	getFirst()	getLast()
pop	getFirst() + removeFirst()	getLast() + removeLast()

Colas

Otra alternativa para trabajar con colas en Java es mediante la clase *Deque* de la librería `java.util.*` la cual posee varias implementaciones, siendo de nuestro interés *LinkedList*.

```
Deque<Integer> cola = new LinkedList<Integer>();
```

De los métodos que contempla, nos interesan:

<code>add(e)</code>	Hace las veces de push: ingresa el elemento <code>e</code> al final de la cola
<code>peek()</code>	Devuelve, más no borra, el elemento al inicio de la cola
<code>poll()</code>	Hace las veces de pop: Devuelve y borra el elemento al inicio de la cola
<code>clear()</code>	Borra todos los elementos de la cola
<code>size()</code>	Devuelve el tamaño de la cola
<code>isEmpty()</code>	Devuelve verdadero si la cola está vacía

Pilas

De igual manera, para las pilas nos sirve la misma clase *Deque* solo que con los siguientes métodos:

```
Deque<Integer> pila = new LinkedList<Integer>();
```

<code>addFirst(e)</code>	Hace las veces de push: ingresa el elemento e al final de la pila
<code>peekFirst()</code>	Devuelve, más no borra, el elemento al inicio de la pila
<code>pollFirst()</code>	Hace las veces de pop: Devuelve y borra el elemento al “tope” de la pila
<code>clear()</code>	Borra todos los elementos de la pila
<code>size()</code>	Devuelve el tamaño de la pila
<code>isEmpty()</code>	Devuelve verdadero si la pila está vacía

Ejemplo

```
import java.util.*;

public class Main {

    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        Deque<Integer> cola = new LinkedList<Integer>();
        Deque<Integer> pila = new LinkedList<Integer>();
        System.out.print("Agregando ");
        for (int i = 10; i <= 50; i += 10) {
            cola.add(i);
            pila.addFirst(i);
            System.out.print(i + " ");
        }
        System.out.println("\nDesencolando");
        for (int j = 0; j < 5; j++) {
            System.out.println(cola.poll());
        }
        System.out.println("Desapilando");
        for (int j = 0; j < 5; j++) {
            System.out.println(pila.pollFirst());
        }
    }
}
```

Atención: todos los ejercicios de esta semana se pueden hacer usando exclusivamente pilas y colas, con sus métodos correspondientes a *push*, *pop*, y *peek* (quizá también se requieran los genéricos *size*, *isEmpty*, y *clear*).

En otras palabras, no es necesario (o incluso no debería) usarse otras cosas como indexación, iteradores, etc.