

Miguel Lázaro Álvarez G77D141

## Práctica 04 Colas -

Ejercicio 1: Operaciones básicas de una cola.

```
class Cola {  
    private int frente;  
    private int fin;  
    private String [] datos;  
  
    public Cola (int capacidad) {  
        datos = new String [capacidad];  
        frente = 0;  
        fin = -1;  
    }  
    public boolean estaVacia () {  
        return fin < frente;  
    }  
    public boolean estaLlena () {  
        return fin == datos.length - 1;  
    }  
    public void enqueue (String elemento) {  
        if (estaLlena ()) {  
            System.out.println ("Cola llena");  
        } else {  
            fin = fin + 1;  
            datos [fin] = elemento;  
        }  
    }  
    public String dequeue () {  
        if (estaVacia ()) {  
            return null;  
        } else {  
            String elemento = datos [frente];  
            frente = frente + 1;  
            return elemento;  
        }  
    }  
}
```

} else {

```
String temp = datos [fronte];  
fronte = fronte + 1;  
return temp;
```

}

}

### Simulación

Operación	Fronte	Fin	Cola
Inicial	0	-1	[]
Enqueue	0	0	[A]
Enqueue	0	1	[A, B]
Enqueue	0	2	[A, B, C]
Enqueue	1	2	[B, C]
Enqueue	1	3	[B, C, D]
Enqueue	1	4	[B, C, D, E]
Enqueue	2	4	[C, D, E]

a) Estado Final: [C, D, E]

b) Elemento en frente: C (posición 2)

c) Elemento en fin: E (posición 4)

Miguel Lozano Arvizu GTID141

## Práctica 04 Colas

### Ejercicio 2: Cola circular

```
class ColaCircular {  
    private Frente;  
    private Fin;  
    private String [] datos;  
    public ColaCircular (int capacidad) {  
        datos = new String [capacidad];  
        frente = 0;  
        fin = 0;  
    }  
    public boolean estaVacia () {  
        return frente == fin;  
    }  
    public boolean estaLlena () {  
        return (fin + 1) % datos.length == frente;  
    }  
    public void enqueue (String elemento) {  
        if (estaLlena ()) {  
            System.out.println ("Cola llena");  
            return;  
        }  
        datos [fin] = elemento;  
        fin = (fin + 1) % datos.length;  
    }  
}
```

```
public String dequeue () {  
    if (estaVacia ()) return null;  
    String temp = datos [frente];  
    frente = (frente + 1) % datos.length;  
    return temp;  
}
```

- Simulacion - capacidad 5 -

Inicial : Frente=0, fin=0, []

Dequeue () Frente=1, fin=0, [] vacina  
no hace nada

Enqueue (A) Frente=1, fin=2, [-, A]

Enqueue (B) Frente=1, fin=3, [-, A, B]

a) Nueva posicion de frente : 1

b) Nueva posicion de fin: 3

c) Si hacemos otro enqueue, se agrega  
en la posicion 3 y fin pasa a 4.

Miguel Lecano Arvizu 6TID141

### Práctica 04 Colas.

Ejercicio 3: Verificar uso correcto de colas (teórico + razonamiento).

a) Personas esperando turno en una ventanilla:

Si es adecuado, porque siguen el principio FIFO, la primera persona en llegar es la primera en ser atendida, lo cual es exactamente el comportamiento de una cola.

b) Impresión de Documentos:

Si es adecuado, porque los documentos se imprimen en el orden en que fueron enviados a la impresora osea el principio FIFO.

c) Deshacer acciones: (UNDO):

No es adecuada, porque deshacer acciones requiere acceder a la última acción realizada osea LIFO, lo que le corresponde a una pila, no a una cola.

d) Reservación de asientos:

Si es adecuado, porque las reservaciones se procesan en orden de llegada osea FIFO, la primera persona en solicitar una reservación es la primera en ser procesada.

Miguel Lozano Arvieu GTID141

## Práctica 04 Colas

### Ejercicio 4. Simulación de colas de proceso.

```
Class Proceso {  
    private String nombre;  
    public Proceso (String n) {  
        nombre = n;  
    }  
    public String getNombre () {  
        return nombre;  
    }  
    public void setNombre (String nombre) {  
        this.nombre = nombre;  
    }  
}  
class ColaProcesos {  
    private int frente;  
    private int fin;  
    private Proceso [] procesos;  
  
    public ColaProcesos (int capacidad) {  
        procesos = new Proceso [capacidad];  
        frente = 0;  
        fin = -1;  
    }  
}
```

```
public void enqueue (Proceso p) {  
    if (fin == procesos.length - 1) {  
        System.out.println ("Cola llena");  
        return;  
    }  
    fin = fin + 1;  
    procesos [fin] = p;  
}
```

```
public Proceso dequeue () {  
    if (fin < frente) {  
        return null;  
    }  
    Proceso temp = procesos [frente];  
    frente = frente + 1;  
    return temp;  
}
```

- Simulación -
- Pregunta: El tercer proceso en ejecutarse es el P3.
- $t=0$ : Enqueue (P1)  $\rightarrow$  Cola: [P1]
  - $t=1$ : Enqueue (P2)  $\rightarrow$  Cola: [P1, P2]
  - $t=2$ : Enqueue (P3)  $\rightarrow$  Cola: [P1, P2, P3]
  - $t=3$ : Dequeue (P1)  $\rightarrow$  Cola: [P2, P3] (P1 ejecutándose).
  - $t=4$ : Enqueue (P4), Dequeue (P2)  $\rightarrow$  Cola: [P3, P4]  
(P2 ejecutándose)

Miguel Lozano Arriwa 6TIO141

Scribel

## Práctica 04 Colas

### Ejercicio 5 - BFS usando una cola

```
class ColaInt {  
    private int [] datos;  
    private int frente;  
    private int fin;  
  
    public ColaInt (int capacidad) {  
        datos = new int [capacidad];  
        frente = 0;  
        fin = -1;  
    }
```

```
    public void enqueue (int x) {  
        fin = fin + 1;  
        datos [fin] = x;  
    }
```

```
    public int dequeue () {  
        int temp = datos [frente];  
        frente = frente + 1;  
        return temp;  
    }
```

```
    public boolean estaVacia () {  
        return fin < frente;  
    }
```

-Simulación BFS (A-B-C-D):

Grafo: 0-1-2-3

Iteración 1: Encolar  $\rightarrow$  Cola: [0]

Iteración 2: Desencolar: (0), Encolar: (1)  $\rightarrow$  Cola [1], Visitado: 0

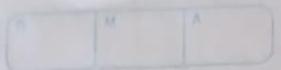
Iteración 3: Desencolar: (1), Encolar: (2)  $\rightarrow$  Cola [2], Visitado: 0, 1

Iteración 4: Desencolar: (2), Encolar: (3)  $\rightarrow$  Cola [3], Visitado: 0, 1, 2

Iteración 5: Desencolar: (3)  $\rightarrow$  Cola [], Visitado: 0, 1, 2, 3

-Orden de vista: 0, 1, 2, 3, (A, B, C, D).

Miguel Lozano Arrieta GTID141



## Práctica 04 Colas.

### Ejercicio 6 - Cola en un Banco.

```
class BancoCola {  
    private int frente;  
    private int fin;  
    private String[] clientes = new String[10];  
  
    public BancoCola(){  
        frente = 0;  
        fin = -1;  
    }  
  
    public void enqueue (String c){  
        fin = fin + 1;  
        clientes [fin] = c;  
    }  
  
    public String dequeue (){  
        String temp = clientes [frente];  
        frente = frente + 1;  
        return temp;  
    }  
}
```

- Simulacion -

- 1.- Enqueue (Carla)  $\rightarrow$  Frente = 0, Fin = 0, Cola: [Carla]
- 2.- Dequeue ()  $\rightarrow$  Frente = 1, Fin = 0, Cola: []
- 3.- Enqueue (Pedro)  $\rightarrow$  Frente = 1, Fin = 1, Cola: [Pedro]
- 4.- Enqueue (Sofia)  $\rightarrow$  Frente = 1, Fin = 2, Cola: [Pedro, Sofia]
- 5.- Dequeue ()  $\rightarrow$  Frente = 2, Fin = 2, Cola: [Sofia]
- 6.- Dequeue ()  $\rightarrow$  Frente = 3, Fin = 2, Cola: []

a) Cola final: [] (vacía)

b) Quien queda al frente: Nadie (vacío)

c) Total personas atendidas: 3 (Carla, Pedro, Sofia)