

Practica 7

Nombre: Juan David Jiménez Romero

Código estático main:

```
package cola;

public class ColaApp {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Cola cola = new Cola(5);

        //insertamos 4 elementos a la cola
        cola.insertar(10);
        cola.insertar(15);
        cola.insertar(20);
        cola.insertar(25);

        while(!cola.colaVacia()) {
            Object n = cola.quitar();
            System.out.print(n);
            System.out.println(" ");
        }

        //insertamos 5 elementos
        cola.insertar("Perro");
        cola.insertar("Gato");
        cola.insertar("Caballo");
        cola.insertar("Leon");
        cola.insertar("Tigre");

        System.out.println("");
        System.out.print("Elemento en la cabeza de la cola: ");
        System.out.print(cola.frenteCola());
        System.out.println("");

        System.out.print("Esta cola esta vacia?");
        System.out.println(cola.colaVacia());

        System.out.print("Esta cola esta llena?");
        System.out.println(cola.colaLlena());

        System.out.print("Tamaño de la cola: ");
        System.out.println(cola.tamanoCola());

        while(!cola.colaVacia()) {
            Object m = cola.quitar();
            System.out.print(m);
            System.out.print(" ");
        }
        System.out.println("");
    }
}
```

Codigo Cola:

```
package cola;
public class Cola {
    private Object[] colaArr;
    //arreglo que contiene los elementos de la cola
    private int tamanoMax;
    //guarda el tamaño maximo del arreglo
    private int frente;
    //apunta al primer elemento de la cola
    private int fin;
    //apunta al ultimo elemento de la cola
    private int numElems;
    //lleva el control del numero de elementos en la cola

    //constructor - inicializa valores de la cola
    public Cola(int tamano) {
        tamanoMax = tamano;
        colaArr = new Object[tamanoMax];
        //instancia arreglo que guarda los elementos de la cola
        fin = -1;
        //la variable fin la inicializamos a -1
        frente = 0;
        //la variable frente la inicializamos a 0
        numElems = 0;
        //numElems se inicializa a 0 (la cola esta vacia, sin elementos)
    }

    //inserta elemento al final de la cola
    public void insertar(Object elemento) {
        if(fin==tamanoMax -1) {
            fin = -1;
            //hacer que la cola sea circular
        }
        fin++;
        //mueve apuntador fin al siguiente indice en colaArr
        colaArr[fin] = elemento;
        //inserta el elemento en la posicion fin del arreglo
        numElems++;
        //aumenta en uno el numero de elementos
    }

    //retira el elemento que se encuentra al frente de la cola

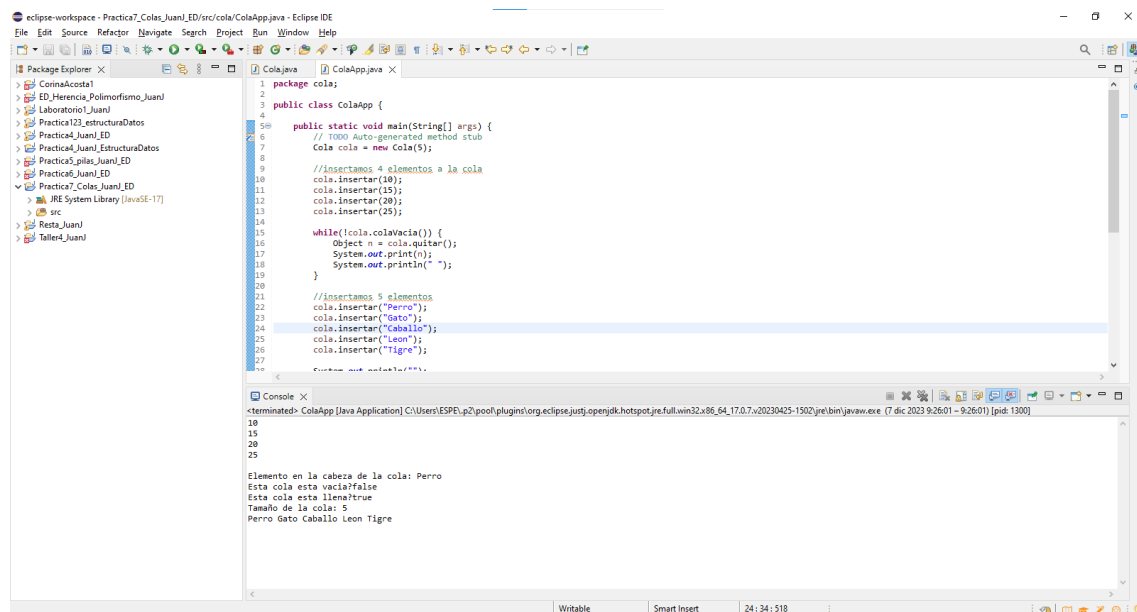
    public Object quitar() {
        Object temp = colaArr[frente];
        //obtiene el elemento del frente de la cola y lo asigna a una
        variable temporal
        frente+=1;
        //se mueve frente una posicion para que apunte al siguiente
        indice de la cola
        if(frente==tamanoMax) {
            frente = 0;
            //hace que la cola sea circular
        }
        numElems--;
        //se reduce en uno el numero de elementos de la cola
        return temp;
    }
}
```

```

        //retormamos el elemento borrador de la cabezade la cola
    }
    //devuelve el elemento que se encuentra al frente o primero en la cola
    public Object frenteCola() {
        return colaArr[frente];
        //retona elemento que esta en la cabeza de la cola
    }
    //devuelve true si la cola esta vacia (no tiene elementos)
    public boolean colaVacia() {
        return (numElems==0);
        //retorna valor booleano dependiendo si la cola esta o no vacia
    }
    //devuelve si la cola esta llena
    public boolean colaLlena() {
        return(numElems==tamanoMax);
        //compara los valores del numero de elementos del arreglo con el
        tamaño del arreglo
    }
    //devuelve el numero de elementos que tiene la cola
    public int tamanoCola() {
        return numElems;
        //retorna valor de la variable numElems
    }
}

```

Ejecución estático:



Código main dinámico:

```

package cola;
import java.util.*;
public class ColaApp {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int opc,a;
        Scanner scan = new Scanner(System.in);
    }
}

```

```

Cola cola = new Cola(5);
int tamano=0;
String animales;

System.out.println("===Menu de opciones===");
System.out.println("1.Agregar datos a la cola");
System.out.println("2.Mostrar datos de la cola");
System.out.println("3.Elemento que esta encima de la cola");
System.out.println("4.Verificar si la cola esta vacia");
System.out.println("5.Verificar si la cola esta llena");
System.out.println("6.Mostrar tamaño de la cola");
System.out.println("7.Vaciar cola");
System.out.println("8.Salir");
opc = scan.nextInt();
while(opc!=8) {
    switch(opc) {
        case 1:
            System.out.println("Ingresa la cantidad de
datos que desees ingresar");
            tamano = scan.nextInt();
            for (int i=0;i<tamano;i++) {
                System.out.println("Ingrese el dato de
la pila N° "+(i+1)+" :");
                a = scan.nextInt();
                cola.insertar(a);
            }
            break;
        case 2:
            for (int j = 0;j<tamano;j++) {
                Object n = cola.frenteCola();
                System.out.print(n);
                System.out.println(" ");
            }
            break;
        case 3:
            System.out.println("");
            System.out.print("Elemento en la
cabeza de la cola: ");
            System.out.print(cola.frenteCola());
            System.out.println("");
            break;
        case 4:
            System.out.print("Esta cola esta
vacía?");
            System.out.println(cola.colaVacia());
            break;
        case 5:
            System.out.print("Esta cola esta
llena?");
            System.out.println(cola.colaLlena());
            break;
        case 6:
            System.out.print("Tamaño de la cola:
");
            System.out.println(cola.tamanoCola());
            break;
        case 7:
            while(!cola.colaVacia()) {
                Object m = cola.quitar();

```

```

        System.out.print(m);
        System.out.println(" ");

    }
    System.out.println("Han sido

eliminados");

    break;
default:
    System.out.println("Opcion no

valida");

    break;

}
System.out.println("===Menu de opciones===");
System.out.println("1.Agregar datos a la cola");
System.out.println("2.Mostrar datos de la cola");
System.out.println("3.Elemento que esta encima de la

cola");

System.out.println("4.Verificar si la cola esta vacia");
System.out.println("5.Verificar si la cola esta llena");
System.out.println("6.Mostrar tamaño de la cola");
System.out.println("7.Vaciar cola");
System.out.println("8.Salir");
opc = scan.nextInt();

}
System.out.println("Gracias por ingresar");

}
}

```

Ejecucion del programa:

```

eclipse-workspace - Practica7_Colas_JuanI_ED/src/cola/ColaApp.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer X
> CorinaAcosta1
> ED_Herencia_Polimorfismo_JuanI
> Laboratorio1_JuanI
> Practica123_estructuraDatos
> Practica4_JuanI_ED
> Practica5_pilas_JuanI_ED
> RE_System_Library (JavaSE-17)
  > src
    > practica
      > GeneraPila.java
      > Nodo.java
      > Pila.java
      > module-info.java
    > Practica6_JuanI_ED
    > Practica7_Colas_JuanI_ED
    > RE_System_Library (JavaSE-17)
    > src
    > Resta_JuanI
    > Taller4_JuanI
ColaApp.java
1 package cola;
2 import java.util.*;

ColaApp [Java Application] C:\Users\ESPE\p2\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7\j2re17\bin\javaw.exe (7 dic 2023 9:44:09) [pid: 7200]

===Menu de opciones===
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
1
Ingrese la cantidad de datos que desees ingresar
4
Ingrese el dato de la pila N° 1 :
10
Ingrese el dato de la pila N° 2 :
15
Ingrese el dato de la pila N° 3 :
20
Ingrese el dato de la pila N° 4 :
25
===Menu de opciones===
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
2
10
10
10
10
===Menu de opciones===
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
3

```

eclipse-workspace - Practica7_Colas_Juan_ED/src/cola/ColaApp.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

ColaApp.java X

```
1 package cola;
2 import java.util.*;
```

Console X

ColaApp [Java Application] C:\Users\ESPE\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (7 dic 2023 9:44:09) [pid: 7200]

```
====Menu de opciones====
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
9

Elemento en la cabeza de la cola: 10
====Menu de opciones====
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
9
10
15
20
25
====Menu de opciones====
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
9
4
Esta cola esta vacia?true
====Menu de opciones====
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
```

eclipse-workspace - Practica7_Colas_Juan_ED/src/cola/ColaApp.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

ColaApp.java X

```
1 package cola;
2 import java.util.*;
```

Console X

ColaApp [Java Application] C:\Users\ESPE\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (7 dic 2023 9:44:09) [pid: 7200]

```
Esta cola esta llena?false
====Menu de opciones====
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
9
1
Ingreso la cantidad de datos que desees ingresar
3
Ingreso el dato de la pila N° 1 :
5
Ingreso el dato de la pila N° 2 :
7
Ingreso el dato de la pila N° 3 :
9
====Menu de opciones====
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
9
6
Tamaño de la cola: 3
====Menu de opciones====
1.Agregar datos a la cola
2.Mostrar datos de la cola
3.Elemento que esta encima de la cola
4.Verificar si la cola esta vacia
5.Verificar si la cola esta llena
6.Mostrar tamaño de la cola
7.Vaciar cola
8.Salir
9
4
Esta cola esta vacia?false
```

