UNIVERSIDAD NACIONAL DE CAJAMARCA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS

ALGORITMOS Y ESTRUCTURA DE DATOS II
10: COLAS

INTRODUCCIÓN

Semana 10 - Colas

COLAS

- Una cola es similar a la fila para pagar en un supermercado: el cajero atiende primero a la persona que se encuentra hasta adelante, y los demás clientes entran a la fila sólo por su parte final y esperan a que se les atienda.

RECORDEMOS

- ¿Qué es una lista enlazada?
- ¿Cuál es la característica principal de una pila?
- ¿En qué se diferencian los procedimientos de una lista enlazada con los de una pila?
- ¿Qué aplicaciones prácticas podemos darle a las pilas?
- ¿Qué entiende por herencia y composición?

PENSEMOS!

- ¿Qué pasaría si dada una cola de personas me piden crear dos coalas basadas en el sexo de una persona, es decir una cola de hombres y una cola de mujeres?

LOGRO ESPERADO

- Al término de la sesión, el estudiante elabora un menú de opciones que gestione una cola de un banco, verificando que se realicen correctamente las tareas principales de una cola.

DESARROLLO DEL TEMA

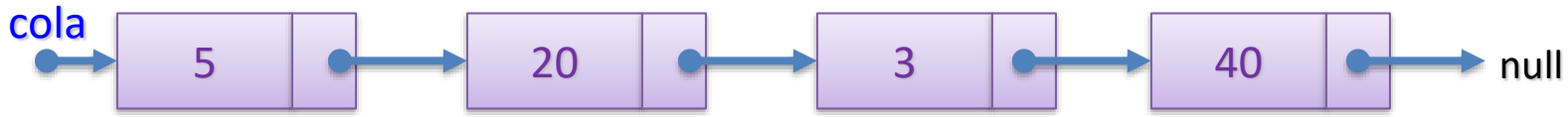
Estructura de Datos

COLAS

- Los nodos de una cola se eliminan sólo desde el inicio (cabeza) de la misma y se insertan sólo al final de ésta. Por esta razón, a una cola se le conoce como estructura de datos PEPS (primero en entrar, primero en salir) o FIFO. Las operaciones para insertar y eliminar se conocen como enqueue (encolar) y dequeue (desencolar).

COLAS

COLAS



Encolar el número 18 en la Cola: Al final

Desencolar un elemento de la Cola (número 5: El primero)

Observe que se inserta al final y se elimina al inicio

IMPLEMENTACIÓN DE UNA COLA: GENÉRICOS

- Para implementar una cola se puede trabajar de manera similar a como se hizo con las Pilas:
  - A través de la herencia, pero se tenía acceso a todos los métodos y
  - Haciendo uso de la composición, es decir definiendo una clase Pila y en ella un atributo de la clase ListaEnalazada, redefiniendo los métodos necesarios para manipular la pila

```
public class Pila {
    private ListaEnlazada unaPila;
}
```

- Los generics fueron introducidos en la versión 5 de Java en 2004 junto con otras muchas novedades.
- Los generics permiten usar tipos para parametrizar las clases, interfaces y métodos al definirlas. Los beneficios son:
  - Comprobación de tipos más fuerte en tiempo de compilación.
  - Eliminación de casts aumentando la legibilidad del código.
  - Posibilidad de implementar algoritmos genéricos, con tipado seguro.

IMPLEMENTACIÓN DE UNA COLA USANDO GENÉRICOS


```
public class Nodo<T> {
   private T dato;
   private Nodo<T> sgt;

   public Nodo(T dato, Nodo<T> sgt) {
      this.dato = dato;
      this.sgt = sgt;
   }
   public Nodo(T dato) {
      this.dato = dato;
      this.sgt = null;
   }
   public T getDato() {
      return dato;
   }
   public void setDato(T dato) {
      this.dato = dato;
   }
   public Nodo<T> getSgt() { return sgt; }

   public void setSgt(Nodo<T> sgt) { this.sgt = sgt; }

   @Override
   public String toString() {
      return dato +"";
   }
}
```

```
public class Cola<T> {
   private Nodo<T> inicio;
   private Nodo<T> fin;
   private int n;

   public boolean estaVacia(){
      return inicio == null;
   }
   public int zise(){
      return n;
   }
}
```

MÉTODOS PRINCIPALES DE UNA COLA

ENCOLAR

```
public void encola(T dato){
    if (estaVacia()){
        inicio = fin = new Nodo<>(dato);
    }else{
        fin.setSgt(new Nodo<>(dato));
        fin = fin.getSgt();
    }
    n++;
}
```

DESENCOLAR

```
public T desencola(){
    if (!estaVacia()){
        T elementoEliminado = inicio.getDato();
        if (inicio == fin){
            inicio = fin = null;
        }else{
            inicio = inicio.getSgt();
        }
        n--;
        return elementoEliminado;
    }
    return null;
}
```

CLASE DE PRUEBA

```
public class pruebaCola {
   public static void main(String[] args) {
       // no se especifica un argumento de tipo
      Cola cola1 = new Cola();
      cola1.encola(1);
      cola1.encola("dos");
      cola1.encola(new Point(3,3));
      cola1.encola(Math.PI);

      System.out.println(cola1);
      while(!cola1.estaVacia()){
         System.out.println("desencolando a ... " +
cola1.desencola());
      }
      System.out.println(cola1);
   }
}
```

En este caso, se dice que “cola1” tiene un tipo crudo, lo cual significa que el compilador utiliza de manera implícita el tipo Object en la clase genérica para cada argumento de tipo.

```
public class pruebaCola {
   public static void main(String[] args) {
      Cola<String> cola2 = new Cola<>();
      cola2.encola("Uno");
      cola2.encola("Dos");
      cola2.encola("Tres");
      cola2.encola("Cuatro");
      cola2.encola("Cinco");

      System.out.println(cola2);
      while(!cola2.estaVacia()){
         System.out.println("desencolando a ... " +
cola2.desencola());
      }
      System.out.println(cola2);
   }
}
```

EVALUACIÓN DEL TEMA DESARROLLADO

Reflexionemos!

RECORDANDO LO APRENDIDO

- ¿Qué es una cola?
- ¿Qué relación existe entre una cola y una lista?
- ¿Qué métodos son necesarios para trabajar con una cola?
- ¿Qué pasaría si se trabaja una cola con una sola referencia al inicio?
- ¿En dónde puede aplicar los conceptos de colas?
- ¿Cuál es la utilidad de los genéricos y en qué diferencia de utilizar el tipo de dato Object?

EJERCICIOS DE APLICACIÓN

EJERCICIO DE APLICACIÓN

- Simule una cola en un banco incluyendo tiempos de atención y espera aleatorios para las diferentes transacciones que se puede realizar y muestre sus resultados