Introducción a Ciencias de la Computación
2020-2
Práctica 4: Listas

Pedro Ulises Cervantes González
confundeme@ciencias.unam.mx

Emmanuel Cruz Hernández
emmanuel_cruzh@ciencias.unam.mx

Adriana Sánchez del Moral
adrisanchez@ciencias.unam.mx

Víctor Zamora Gutiérrez
agua@ciencias.unam.mx

Fecha límite de entrega: 24 de abril de 2020.
Hora límite de entrega: 23:59.

1. Objetivo

Aprender a programar el comportamiento de una lista, así como conocer su diseño, para un entendimiento más profundo de su funcionamiento.

Además, de conocer algunas aplicaciones de las listas en la vida real para almacenar información y modificarla en tiempo de ejecución de un programa.

2. Actividades

Crea una clase llamada *ListaLigada* que implemente la interfaz *ListaInterfaz*.

2.1. Actividad 1 *(0.5 puntos)*

Implementa el método *tamanio*, que permite saber el tamaño de la lista.

2.2. Actividad 2 *(1 punto)*

Implementa el método *cola* que devuelve la cola de la lista. Esto es, la lista sin el primer elemento.

2.3. Actividad 3 *(0.5 puntos)*

Implementa el método *insertaPrimero* que permite insertar un elemento al principio de la lista.

2.4. Actividad 4 *(1 punto)*

Implementa el método *inserta* que permite insertar un elemento en una posición específica de la lista.

2.5. Actividad 5 *(1 punto)*

Implementa el método *insertaFinal* que permite insertar un elemento en la última posición de la lista.

2.6. Actividad 6 *(0.5 puntos)*

Implementa el método *obtenPrimero* que permite acceder al primer elemento contenido en una lista.

2.7. Actividad 7 *(1 punto)*

Implementa el método *obten* que permite acceder a un elemento en una posición específica en una lista.

2.8. Actividad 8 *(0.5 puntos)*

Implementa el método *eliminarPrimero* que permite eliminar el primer elemento de una lista.

2.9. Actividad 9 *(1 punto)*

Implementa el método *elimina* que permite eliminar un elemento en una posición específica de una lista.

2.10. Actividad 10 *(1 punto)*

Implementa el método *indice* que permite buscar un elemento en la lista y devuelve su primera aparición.

2.11. Actividad 11 *(0.5 puntos)*

Implementa el método *estaVacia* que permite saber si una lista tiene o no elementos.

2.12. Actividad 12 *(1 punto)*

Implementa el método *concatena* que permite agregar los elementos de la lista invocante con una lista nueva.

2.13. Actividad 13 *(0.5 puntos)*

Implementa el método *limpia* que elimina todos los elementos de una lista.

3. Nota importante

Pon a prueba todos los métodos de la práctica en el método main de la clase *Lista*. Imprime en terminal las entradas y los resultados de cada método. **Usar el Scanner es opcional**.

También debes implementar un método que imprima todos los elementos de la lista. Por cada operación que pongas a prueba, debes mostrar las modificaciones de la lista.

4. Punto Extra

Implementa todos los métodos de la interfaz *ListaOrdenadaInterfaz* en una clase llamada *ListaOrdenada*, que cuenta con la característica de tener todos los elementos contenidos ordenados.

Pon a prueba todos los métodos del punto extra en el método main. Imprime en terminal las entradas y los resultados de cada método.

NOTA: puedes hacer la lista ordenada de elementos tipo *int*, *double*, *char*, etc. Para poder tener un orden en el contenido de la lista.

El formato de entrega del punto extra es el siguiente:

- *Practica04ParternoNombre*
  - src
    - ListaInterfaz.java
    - Lista.java
    - Nodo.java
  - extra
    - ListaOrdenadaInterfaz.java
    - ListaOrdenada.java
    - Nodo.java
  - Readme.txt

5. Créditos

Interfaces creadas por Lic. en Ciencias Computacionales Pedro Ulises Cervantes González para el curso de Introducción a las Ciencias de la Computación 2017-2.

6. Materiales para consultar

1. `https://docs.google.com/presentation/d/1T0DBW4FUK4EA5w98H8c0ZerQjavJfoULE9G1U5dnrYE/edit?usp=sharing`
2. `https://youtu.be/lyqRbcxhXY4`
3. `https://www.youtube.com/watch?v=VoPRnfn06VI`
4. `https://youtu.be/hIpW1Djw1Qk`

7. Reglas Importantes

- Cumple con los lineamientos de entrega.
- Todos los archivos deberán contener nombre y número de cuenta.
- Tu código debe estar comentado. Esto abarca clases, atributos, métodos y comentarios extra.
- Para cada clase solicitada, crea un nuevo archivo.
- Utiliza correctamente las convenciones para nombrar variables, constantes, clases y métodos.
- Sólo se permite el uso de la biblioteca Scanner.
- **SÍ** se reciben prácticas con retraso. Por cada día se restará 1 punto.
- En caso de no cumplirse alguna de las reglas especificadas, se restará 0.5 puntos en tu calificación obtenida.