Informática II Nivel Superior  
Prepa Tec Campus Eugenio Garza Lagüera  
**Laboratorio Final**

1. **Repasa los siguientes conceptos. ¿Qué significan, y cómo se aplican en la programación orientada a objetos?**
   1. Clases y objetos
      * Constructores
   2. Variables de instancia y variables de clase
   3. Encapsulación
      * Modificadores de acceso privado, público, protected, default
      * Métodos setters y getters
      * Uso de palabra reservada *this*
      * Information hiding
   4. Métodos de instancia y métodos estáticos
      * Sobrecarga de métodos (overloading)
   5. Variables primitivas y variables de tipo referencia
   6. Arreglos bidimensionales y arreglos multidimensionales
      * Declaración de arreglos
      * Leer / modificar contenido de arreglos
      * Enviar arreglos como parámetro en métodos
   7. Algoritmos de ordenamiento y búsqueda:
      * Búsqueda secuencial
      * Búsqueda binaria
      * Bubble Sort
      * Selection Sort
      * Merge Sort
   8. Recursión
   9. Análisis de Algoritmos
      * Notación Big-O
      * Comparación de algoritmos
   10. Herencia y polimorfismo
       * Uso, ventajas y desventajas
       * Clases base y clases derivadas
       * Redefinición de métodos (overriding)
       * Esconder métodos (hiding)
       * Método super( )
   11. Estructuras de datos dinámicas
       * Hashtable, Stack, Queue, Listas encadenadas
       * Operaciones (agregar, eliminar, buscar, recorrer).
       * Ventajas y desventajas
   12. Archivos
       * Lectura de archivos
       * Creación de archivos
       * Escritura sobre archivos

**Reporte Becario**

**Sección 1.** En servicio becario, el director de la Prepa Tec te ha pedido que le prepares un reporte con los temas que más les preocupan a los alumnos, por lo que te ha dado acceso a la cuenta de Instagram de la @prepatecmty. Además, te ha solicitado que elimines todos los nombres del archivo, para que el reporte sea 100% anónimo.

Diseña una clase llamada **StudentComments** con los siguientes atributos, recordando utilizar las mejores prácticas de la programación orientada a objetos.

* Variable **File report** que contenga la referencia a un archivo del sistema operativo.
* Variable **contents** de tipo **LinkedList<String>** que sirva para almacenar el contenido de todos los comentarios recopilados de los alumnos.
* Método constructor **StudentComments(String filepath)** que instancie el objeto con la lista **contents** vacía, y el Objeto **report** construido con el **filepath** recibido.
* Método **isValidFile()** que retornará **true** cuando el objeto **report** apunte a un archivo existente. Utiliza el método **exists()** de la clase **File**.
* Método **void replaceInComments(String from, String to)** que itere sobre la lista **contents** y sustituya todas las ocurrencias del **String** **from** por el **String to.**
* Método **void addComment(String comment)** que agregue el contenido del parámetro de entrada **comment** al final de la lista **contents**.
* Método **void removeComment(int line**) que elimine de la lista **contents** la línea recibida como parámetro de entrada.
* Método **boolean saveFile()** que escriba el contenido de la variable **contents** en el archivo **report**. El método retornará **true** si la operación pudo ser terminada satisfactoriamente.
* Método **LinkedList<String> readFile()** que lea el archivo al que apunta el objeto report, y retorne una lista con todas las líneas que dicho archivo contenga.

OJO! El método **saveFile()** deberá ser explícitamente llamado después de cada operación **addComment()**, **removeComment()** y **replaceInComments()** para que el contenido del archivo se actualice.

**Sección 2.** Diseña adicionalmente una interfaz gráfica para el programa solicitado, que permita realizar las siguientes funciones:

A screenshot of a cell phone

Description automatically generated

* El programa tendrá una pantalla de bienvenida que te permita ingresar la ubicación y ruta de un archivo de texto en donde se generará el reporte.
* Al presionar el botón “Valid file?”, el programa instanciará un objeto de la clase **StudentComments** (sugerencia, la clase MainWindow puede tener una variable de instancia que sea un objeto de la clase **StudentComments).** Si hay algún error, deberá mostrar en un label la leyenda “Archivo inválido”. La persona no podrá hacer uso de los botones “Delete Comment”, “Update Comment” o “Add Comment” hasta que se corrija este error.
* Agrega funcionalidad al programa para que el usuario pueda:
  + Agregar un comentario al final del archivo
  + Eliminar un comentario existente mediante su índice
  + Modificar el contenido de todo el archivo, reemplazando un String por otro.

**Sección 3.** Resuelve por lo menos 7 problemas de la lista de abajo. Elige 1 problema de cada tema, y resuélvelo de acuerdo con los comentarios. Al finalizar, genera un archivo de java que contenga todos los métodos generados.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID Leetcode** | **Link** | **Tema** | **Comentario** |
| 1 | <https://leetcode.com/problems/two-sum/> | Hashtables |  |
| 20 | <https://leetcode.com/problems/valid-parentheses/> | Stacks |  |
| 27 | <https://leetcode.com/problems/remove-element/> | Arrays |  |
| 35 | <https://leetcode.com/problems/search-insert-position/> | Binary Search |  |
| 118 | <https://leetcode.com/problems/pascals-triangle/> | Listas |  |
| 136 | <https://leetcode.com/problems/single-number/> | Hashtables | Pueden resolverlo utilizando memoria extra. |
| 167 | <https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/> | Binary Search | El algoritmo debe ser O(n log(n) ). No quiero fuerza bruta. |
| 202 | <https://leetcode.com/problems/happy-number/> | Hashtables |  |
| 242 | <https://leetcode.com/problems/valid-anagram/> | Sorting | Construir una solución utilizando MergeSort. No usar el método Arrays.sort(). |
| 258 | <https://leetcode.com/problems/add-digits/> | Recursión | Realizar la suma de todos los dígitos con un algoritmo recursivo. |
| 344 | <https://leetcode.com/problems/reverse-string/> | Recursión | Resolver utilizando recursión. |
| 349 | <https://leetcode.com/problems/intersection-of-two-arrays/> | Hashtables |  |
| 844 | <https://leetcode.com/problems/backspace-string-compare/> | Stacks |  |
| 1356 | <https://leetcode.com/problems/sort-integers-by-the-number-of-1-bits/> | Sorting | Construyan un método que convierta un número entero a binario. |