Informática II - Prepa Tec Campus Eugenio Garza Lagüera  
Laboratorio Primer Parcial

**Sección 1:** Diseña una clase llamada CreditCard que sirva para modelar una tarjeta de crédito. Enlista por lo menos 5 atributos (con sus respectivos tipos de datos) y 4 métodos. Incluye una descripción corta de lo que haría cada método.

**Sección 2:** Utilizando la clase CreditCard diseñada en ejercicio anterior, modela tres objetos distintos de esta clase, con valores únicos para cada uno de los atributos.

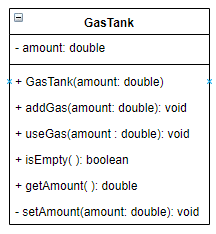
**Sección 3:** Ordena los siguientes arreglos a mano ascendentemente, mostrando cada uno de los pasos, utilizando los algoritmos de Selection Sort y Bubble Sort.

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 4 | 3 | 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 3 | 3 | 4 | 5 |

**Sección 3: Resuelve los siguientes programas en tu computadora. Al finalizar, sube a Blackboard los archivos de código fuente (.java).**

**Problema 1:** Diseña una clase **GasTank** que sirva para representar un tanque de gasolina. Diseña la clase de acuerdo con el siguiente diagrama UML:



* **Constructor:** Este método debe recibir como parámetro de entrada una cantidad **amount**, y llame al método **setAmount** para actualizar la variable de instancia **amount.**
* **addGas**:deberá incrementar la cantidad de gasolina en el tanque en la cantidad recibida como parámetro**.** Asegúrate de validar que sólo se procesen valores positivos.
* **useGas:** deberá reducir la cantidad de gasolina en el tanque en la cantidad recibida como parámetro. Asegúrate de que sólo se procesen valores positivos.
* **isEmpty:** Deberá devolver **true** cuando la cantidad de gasolina en el tanque sea menor a 0.1. De lo contrario, deberá retornar **false.**
* **getAmount:** Getter para la variable amount.
* **setAmount:** Método privado (sólo será usado por el constructor) que actualice la variable **amount** siempre y cuando el parámetro recibido sea mayor o igual a 0.

**Problema 2:** Escribe un método estático **char[ ] removeDuplicates(char[ ] in)** que retorne un nuevo arreglo de caracteres sin caracteres duplicados. Siempre deberás mantener el primer elemento encontrado y eliminar los elementos subsecuentes.  
Ejemplo:  
removeDuplicates(new char[]{'b','d','a','b’,’f’,’a’,’g’,’a’,’a’} ) 🡪 {‘b’,’d’,’a’,’f’,’g’}.