**Programación orientada a objetos**

**Ejercicio elaboración de pruebas unitarias**

**Usted fue contratado como ingeniero de automatización d pruebas de la empresa Protech RaizesTech. Esta empresa tiene un software que fue elaborado para la gestión de cobros y descuentos a propietarios de la unidad.**

**Existen dos tipos de propietarios. Propietarios fundadores y propietarios de reventa. Todo propietario tiene nombre, identificación y una única propiedad. Cada propiedad tiene un número de piso, un número de identificación, un área. Algunas propiedades tienen parqueadero y otras no.**

**Los propietarios fundadores son aquellos que compraron el apartamento directamente a la constructora. Los propietarios de reventa son aquellos propietarios que compraron su propiedad a antiguos propietarios.**

**Los propietarios fundadores tienen derecho a usar el salón social de manera gratuita.**

**Los propietarios fundadores tienen un descuento del 1% para la compra de productos en la tienda del edificio. Los propietarios de reventa tienen un descuento el 0,05 en la mensualidad del gimnasio.**

**Para cada propiedad el propietario debe pagar la administración teniendo en cuenta lo siguiente:**

**\* Cobro por ascensor. Vale 2000 pesos que se multiplican por el piso en el que se encuentre el**

**apartamento.**

**\* El valor base. Cada apartamento para el año 2021 paga 50 mil pesos. La tarifa podría cambiar cada año.**

**\* Area. Hay un recargo del 5 por ciento sobre el valor base para los apartamentos de mas de 150mts**

**La administración quiere:**

* **Conocer el valor a pagar por administración dado la identificación del propietario.**
* **CambiarTipoPropietario. Ocurre cuando un propietario fundador le vende a un propietario de reventa. En esos casos se elimina el propietario fundador y se crea un nuevo propietario de reventa con los datos del nuevo propietario.**
* **Imprimir para cada propietario los beneficios a los que tiene derecho. En particular el administrador quiere una cadena de texto que indique los beneficios.**
* **Ver la lista de propietarios de propiedades que tienen parqueadero.**
* **Conocer el valor recaudado por administración para todo el edificio.**

**Se pide:**

**Observe:**

* **Observe los constructores de la clase Propietario, Reventa y Fundador. Note que los constructores de las clases hijas llaman al constructor de la clase padre para aprovechar el reuso del código. De esta manera se evita que el mismo código se encuentre duplicado.**
* **La prueba elaborada en PropietarioTest.cpp. Así mismo podrá encontrar ejemplos de pruebas en el repositorio en la siguiente ruta https://github.com/lufe089/POO/tree/main/UnitTesting-1**

**Para facilitar las pruebas:**

**Cambie los métodos que imprimen los valores en consola para que retornen los valores esperados. Por ejemplo, si el método imprime un mensaje en lugar de imprimirlo en consola guárdelo en una cadena que retorna. Esta cadena le servirá para comparar los valores retornados con los valores esperados.**

**Para mejorar:**

**Agregue manejo de excepciones para el caso en el que no exista el propietario para las funcionales de mostrarBeneficios y cambiarTipoPropiedades.**

**Implementar las pruebas para:**

* **El flujo básico de las cinco funcionalidades propuestas**
* **El flujo de excepción de mostrarBeneficios y cambiarTipoPropiedades**

**Corrija los errores que encuentre en el código una vez ejecute las pruebas.**

**Para correr las pruebas unitarias desde el IDE:**

1. **Click derecho en CMakeList del proyecto->reload cmake project. El que se encuentra por fuera de las carpetas.**
2. **Espere a que se compile el proyecto y se genere una carpeta llamada cmake-build-debug**

Graphical user interface, text

Description automatically generated with medium confidence

**Revise que no tenga errores en la vista de terminal y en la vista Cmake del CLion**

**Si no tiene errores de click derecho sobre el archivo de pruebas y seleccione: “Run ALL in”**

A screenshot of a computer

Description automatically generated with medium confidence

**Los resultados de la prueba deben pasar.**

Graphical user interface, text, application, email

Description automatically generated

**Para correr las pruebas unitarias por consola**

**Desde el terminal de CLION u otro terminal ingrese a la carpeta del proyecto creada por CMAKE**

**Cmake-build-debug**

Graphical user interface, text

Description automatically generated

Busque entro de la carpeta test el archivo niza\_full\_test.exe y ejecútela en consola

Text

Description automatically generated

***Errores de configuración posibles:***

***Falta comentar el archivo main.***

Graphical user interface, text, application, email

Description automatically generated

**Comenté la lógica que se encuentra en el archivo main para que la suite de pruebas encuentre el main por defecto de la suite de pruebas de Google test.**