

TSU Software, Materia: Programación, Clave 50086, Semestre 16-P
Practica 09. Arreglos de Clases.

1. Objetivos

Los objetivos buscados en esta práctica son los siguientes:

- Identificar atributos, métodos y composición de clases
- Escribir clases y clases de prueba.

2. Ejercicios.

2.1 Ejercicio 1

Desarrollar la clase Usuario que tenga los siguientes atributos:

- No. de cuenta.
- Nombre del usuario.
- Contraseña.
- Grupo.

Proporcionar a la clase tres diferentes constructores, usted decida la lógica de programación más adecuada para cada constructor (revise el ejemplo de la clase Fecha desarrollado en el salón).

Proporciones métodos individuales para asignar y obtener los valores de cada atributo (*getters and setters*).

2.2 Ejercicio 2

Desarrolle una clase de prueba para la clase Usuario, donde compruebe el correcto funcionamiento de su implementación, usando cada uno de los constructores y métodos desarrollados.

2.3 Ejercicio 3

Desarrollar una clase RegistroUsuarios, que tenga como atributo:

- Lista de usuarios.

Esta lista de usuarios será un arreglo de objetos, por lo cual deberá implementar un constructor para inicializar este tipo de dato referenciado (Revise el ejercicio desarrollado en clase Directorio). Además de su constructor esta clase implementara los siguientes métodos:

Método iniciarSesion

Este método pide un número de cuenta y una contraseña. Después, compara el número de cuenta con los que existen en una lista de usuarios. De no existir, informa al usuario que el usuario no existe. Si existe el usuario, compara la contraseña introducida por el usuario con la contraseña almacenada en el objeto que fue encontrado. Si estos dos datos coinciden, da un mensaje diciendo que se ha iniciado sesión correctamente. De lo contrario, da un error de autenticación.

Método de crearUsuario

Este método pide al usuario un número de cuenta, un nombre, una contraseña y un grupo a elegir de una lista de grupos disponibles. El método deberá pedir también que el usuario confirme que la contraseña introducida sea correcta (esto se hará haciendo que el usuario introduzca la contraseña dos veces). Después de haber recibido los datos, comparará el no de cuenta recibido con el de los usuarios existentes. Si existe, arroja un error informando que el usuario ya existe y pedirá cambiar el no de cuenta.

Si el usuario no existe entre los usuarios previamente creados, almacena el usuario en la lista de usuarios existentes e inicia sesión utilizando el no de cuenta y la contraseña otorgada en la creación del usuario como datos de entrada.

2.4 Ejercicio 4

Desarrolle una clase de prueba para la clase RegistroUsuarios, donde compruebe el correcto funcionamiento de su implementación: desde esta misma prueba deberá inicializar los registros de cada usuario, para después hacer uso del método **iniciarSesion** y comprobar el acceso de un Usuario.