Ejercicio 1

Implementar una aplicación para la gestión de un taller

- Clase Vehiculo:
  - Matricula:
    - Campo único
    - Clase formada por 4 digitos – 3 letras
      - Comprobar que la matricula es correcta
  - Modelo
  - KM
- Clase Taller
  - TablaHash Que almacena el conjunto de vehículos
    - Clave : Matricula
  - Metodos :
    - void añadirVehiculo(Vehiculo v)
    - void mostrarVehiculos()
    - Vehiculo buscarVehiculo(String matricula)

Ejercicio 2

Implementar una agenda telefónica

- Clase telefono
  - String de 9 digitos
    - Comprobar que el telefono tiene un formato correcto
- Clase contacto
  - String nombre
  - Teléfono Telefono
  - String dirección
- Clase agenda
  - Tabla hash que almacena un conjunto de contactos
    - Clave teléfono
  - Metodos
    - boolean esVacia()

- void nuevaEntrada (String numero, String nombre, String dirección)
- Contacto buscaContactoTelefono(String numero)
- Contacto buscaContactoNombre(String nombre)
- void mostrarAgenda()
- void imprimirAgenda()

Ejercicio 3

Como parte de una aplicación queremos implementar un modulo para la comprobación de un login de usuario.

Un usuario esta compuesto por un nombre de ususario y una clave

- El nombre de usuario debe de cumplir las siguientes condiciones
  - Tener una longitud entre 4 y 6 caracteres
  - Solo ppuede contener caracteres
- Contraseña
  - Debe de tener una longitud entre 6 y 10 caracteres
  - Debe contener al menos un carácter mayuscula y un número

Cuando un usuario se logea introduce su nombre de ususario y su clave.

El sistema realiza las siguientes comprobaciones:

1. Comprueba que el nombre de usuario se encuentra almacenado
- Si el usuario no está registro muestra un mensaje de advertencia, invitando al usuario a darse de alta
2. Si el ususario está registrado comprueba que el valor de la contraseña coincide
- Si no coincide muestra un mensaje de error, indicando que la contraseña no es correcta

El conjunto de usuarios se almacena en una tabla hash usando como clave el nombre de usuario

La clase login deberá tener los siguientes métodos:

- void nuevoRegistro(String nombreUsuario, String contraseña)
- void login(String nombreUsuario, String contraseña)
- void modificarContraseña(String nombreUsuario, String contraseñaActual, String contraseñaNueva)

Ejercicio 4

Se dispone de una aplicación de control del número de veces que una dirección web ha sido visitada. Para ello utilizaremos una tabla hash <DireccionHTTP, Integer>

Una dirección HTTP se compone de los siguientes campos

- String servidor
- int puerto
- String dirección

Ejemplo: una dirección http://www.host.com:80/datos/fichero.txt es formada por un servidor (www.host.com) , un puerto (80) y una dirección al recurso solicitado (/datos/fichero.txt)

Se pide crear una aplicación capaz recibir un String, descomponer una dirección HTTP en los 3 campos e incrementar el número de accesos

Ejercicio 5

Queremos implementar una aplicación para la gestión de un taller

- Vehiculo
  - Matricula: String
  - Modelo: : String
- Cliente:
  - Dni: : String
  - Nombre : : String
- Reparacion :
  - Id : autonumérico
  - Matricula: : String

- Dni : : String
- Precio : double

En la clase taller almacenamos las siguientes estructuras:

- hashMapa<,dni,Cliente>
- hashMap<matricula,Vechiculo>
- List<Reparacion>

Para poder isnertar una nueva reparación es necsario garantizar quel a información del cliente y el vehiculo se encuentran en las estructuras

Se pide :

- InsertarVehiculo(Vehiculo v)
- insertarCliente(Cliente c)
- insertarReparacion(String dni, String matricula)
- mostrarDatos()
- reparacionesPorCliente(String dni)
- repacacionesPorVehiculo(String matricula)

Ejercicio 2