**Clase Paseo**

/\*\*

\* Nombre del metodo: Paseo (constructor)<br />

\* <br />

\* Descripcion: este metodo se encarga de crear un nuevo objeto Paseo.

\* Inicializa las relaciones<br />

\*

\*/

/\*\*

\*

\* Nombre del metodo: registrarEstudiante<br />

\* <br />

\* Descripcion: Este metodo se encarga de registrar un nuevo estudiante y

\* guardar su referencia en el arraylist de estudiantes<br />

\* <b>Pre: </b> el arraylist de estudiantes debe estar inicializado. <br />

\* <b>Post: </b> el arraylist de estudiantes debe aumentar su tamano en una

\* unidad

\*

\* **@param** numeroIdentificacion

\* - String que representa el numero de identificacion del

\* estudiante. numeroIdentificacion != null, numeroIdentificacion

\* != null

\* **@param** tipoIdentificacion

\* - String que representa el tipo de identificacion del

\* estudiante. . tipoIdentificacion ==

\* Estudiante.TIPO\_IDENTIFCACION\_CC || tipoIdentificacion ==

\* Estudiante.TIPO\_IDENTIFCACION\_CC

\* **@param** nombre

\* - String que representa el nombre del estudiante.

\* nombre != null, nombre != ""

\* **@param** apellidos

\* - String que representa el nombre del estudiante.

\* apellidos != null, apellidos != ""

\* **@param** edad

\* - int que representa la edad del estudiante. edad

\* > 15

\* **@throws** Exception

\* - lanza una excepcion si ya existe un estudiante con el mismo

\* tipo y numero de identificacion

\*/

/\*\*

\*

\* Nombre del metodo: buscarEstudiante (opcional)<br />

\* <br />

\* Descripcion: Este metodo opcional se encarga de buscar un estudiante en el

\* arraylist de estudiantes. La busqueda se basa en el tipo y numero de

\* identificacion<br />

\*

\* **@param** numeroIdentificacion

\* - String que representa el numero de identificacion del

\* estudiante. numeroIdentificacion != null, numeroIdentificacion

\* != null

\* **@param** tipoIdentificacion

\* - String que representa el tipo de identificacion del

\* estudiante. . tipoIdentificacion ==

\* Estudiante.TIPO\_IDENTIFCACION\_CC || tipoIdentificacion ==

\* Estudiante.TIPO\_IDENTIFCACION\_CC

\* **@return** Estudiante - retorna un objeto de tipo Estudiante si encuentra un

\* estudiante en el arraylist con el tipo y numero de identificacion

\* dados. Retorna null en caso contrario.

\*/

/\*\*

\* Nombre del metodo: generarReporteEstudiantesAlojados<br />

\* <br />

\* Descripcion: Este metodo se encarga de generar una cadena de texto que

\* contiene la informacion de los estudiantes que estan registrados en algun

\* alojamiento. La cadena debe seguir el siguiente formato<br />

\* Estudiantes inscritos<br />

\* <br />

\* <br />

\* 1. CC - 20030123, Pepito Perez Mendoza, Alojamiento 1001<br />

\* <br />

\* 2. TI - 20030124, Pepita Perez Ruiz, Alojamiento 1002<br />

\* <br />

\* <br />

\* <br />

\* Si no hay estudiantes registrados aun, entonces se debe mostrar: <br />

\* <br />

\* No hay estudiantes con alojamiento

\* <b>Pre: </b> el arraylist de vehiculos

\* debe estar inicializado. <br />

\*

\* **@return** String - cadena que contiene la informacion sobre los estudiantes

\* alojados

\*/

/\*\*

\* Nombre del metodo: registrarVehiculo<br />

\* <br />

\* Descripcion: Este metodo se encarga de registrar un nuevo vehiculo y

\* guardar su referencia en el arraylist de vehiculos<br />

\* <b>Pre: </b> el arraylist de vehiculos debe estar inicializado. <br />

\* <b>Post: </b> el arraylist de vehiculos debe aumentar su tamano en una

\* unidad

\*

\* **@param** placa

\* - String que corresponde a la placa del vehiculo

\* **@param** tipoVehiculo

\* - String que indica el tipo del vehiculo

\* **@throws** Exception

\* - Si el vehiculo ya se encuentra registrado

\*/

/\*\*

\* Nombre del metodo: registrarAlojamiento<br />

\* <br />

\* Descripcion: Este metodo se encarga de registrar un nuevo alojamiento y

\* guardar su referencia en el arraylist de alojamientos<br />

\* <b>Pre: </b> el arraylist de alojamientos debe estar inicializado. <br />

\* <b>Post: </b> el arraylist de alojamientos debe aumentar su tamano en una

\* unidad

\*

\* **@param** identificador

\* - String que corresponde al identificador del alojamiento

\* **@throws** Exception

\* - Si el alojamiento ya se encuentra registrado

\*/

/\*\*

\* Nombre del metodo: calcularCostoTotalAlojamiento<br />

\* <br />

\* Descripcion: Este metodo se encarga de calcular el costo total en

\* alojamiento. Solo se tienen en cuenta los alojamientos que tienen

\* asignados algun estudiante. Recuerde utilizar la constante para el

\* precio de galon de gasolina<br />

\* <b>Pre: </b> el arraylist de alojamientos debe estar inicializado. <br />

\*

\* **@return** double que corresponde al valor total que se debe pagar en el

\* paseo por concepto de alojamiento para los estudiantes

\*/

/\*\*

\* Nombre del metodo: calcularCostoTotalGasolina<br />

\* <br />

\* Descripcion: Este metodo se encarga de calcular el costo total en

\* gasolina. Solo se tienen en cuenta los vehiculos que tienen asignado un

\* alojamiento<br />

\* <b>Pre: </b> el arraylist de vehiculos debe estar inicializado. <br />

\*

\* **@return** double que corresponde al valor total que se debe pagar en el

\* paseo por concepto de combustible para los vehiculos

\*/

/\*\*

\* Nombre del metodo: asignarAlojamientoVehiculo<br />

\* <br />

\* Descripcion: Este metodo se encarga de buscar un vehiculo por la placa,

\* buscar un alojamiento por el identificador y asignar este ultimo al

\* vehiculo encontrado<br />

\*

\* \* <b>Pre: </b> el arraylist de vehiculos debe estar inicializado. <br />

\* \* <b>Pre: </b> el arraylist de alojamientos debe estar inicializado. <br />

\*

\* **@param** identificadorAlojamiento

\* - String que corresponde al identificador del alojamiento

\* el identificador del vehiculo existe

\* **@param** placaVehiculo

\* - String que corresponde a la placa del vehiculo

\* la placa del vehiculo existe

\* **@return** boolean - true si el vehiculo no tenia alojamiento asignado -

\* false en caso contrario

\*/

/\*\*

\* Nombre del metodo: alojarEstudiante<br />

\* <br />

\* Descripcion: Este metodo se encarga de buscar un alojamiento dado el

\* identificador, buscar un estudiante por tipo y numero de identificacion y

\* albergar este ultimo en el alojamiento encontrado<br />

\*

\* <b>Pre: </b> el arraylist de alojamientos debe estar inicializado. <br />

\* <b>Pre: </b> el arraylist de estudiantes debe estar inicializado. <br />

\* <b>Pre: </b> el estudiante no ha sido alojado aun.

\*

\* **@param** identificadorAlojamiento

\* - String que corresponde al identificador del alojamiento

\* El identificador existe

\* **@param** tipoIdentificacion

\* - String que corresponde al tipo de identificacion <b>Pre:

\* </b> El tipo existe

\* **@param** numeroIdentificacion

\* - String que corresponde al numero de identificacion <b>Pre:

\* </b> El numero existe

\*

\* **@return** boolean - true si se puede albergar al estudiante, false en caso

\* contrario

\*

\*/

/\*\*

\* Nombre del metodo: darEstudiantes<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor de la relacion

\* estudiantes.<br />

\*

\* **@return** ArrayList<Estudiante> que representa la relacion con los

\* estudiantes

\*/

/\*\*

\* Nombre del metodo: darVehiculos<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor de la relacion

\* vehiculos.<br />

\*

\* **@return** ArrayList<Vehiculo> que representa la relacion con los

\* vehiculos

\*/

/\*\*

\* Nombre del metodo: darAlojamientos<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor de la relacion

\* alojamientos.<br />

\*

\* **@return** ArrayList<Alojamiento> que representa la relacion con los

\* alojamientos

\*/

**Clase Estudiante**

\* Nombre del metodo: Estudiante (constructor)<br />

\* <br />

\* Descripcion: este metodo se encarga de crear un nuevo objeto Estudiante.

\* Inicializa los atributos con los valores que vienen como parametro<br />

\*

\* **@param** numeroIdentificacion

\* - String que representa el numero de identificacion del

\* estudiante. numeroIdentificacion != null, numeroIdentificacion

\* != null

\* **@param** tipoIdentificacion

\* - String que representa el tipo de identificacion del

\* estudiante. . tipoIdentificacion ==

\* Estudiante.TIPO\_IDENTIFCACION\_CC || tipoIdentificacion ==

\* Estudiante.TIPO\_IDENTIFCACION\_CC

\* **@param** nombre

\* - String que representa el nombre del estudiante.

\* nombre != null, nombre != ""

\* **@param** apellidos

\* - String que representa el nombre del estudiante.

\* apellidos != null, apellidos != ""

\* **@param** edad

\* - int que representa la edad del estudiante. edad

\* > 15

\*/

/\*\*

\* Nombre del metodo: darNumeroIdentificacion<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* numeroIdentificacion.<br />

\*

\* **@return** String que contiene el valor del atributo numeroIdentificacion

\*/

/\*\*

\* Nombre del metodo: darTipoIdentificacion<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* tipoIdentificacion.<br />

\*

\* **@return** String que contiene el valor del atributo tipoIdentificacion

\*/

/\*\*

\* Nombre del metodo: darNombre<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* nombre.<br />

\*

\* **@return** String que contiene el valor del atributo nombre

\*/

/\*\*

\* Nombre del metodo: darApellidos<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* apellidos.<br />

\*

\* **@return** String que contiene el valor del atributo apellidos

\*/

/\*\*

\* Nombre del metodo: darEdad<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* edad.<br />

\*

\* **@return** int que contiene el valor del atributo edad

\*/

**Clase Alojamiento**

/\*\*

\* Nombre del metodo: Alojamiento (constructor)<br />

\* <br />

\* Descripcion: este metodo se encarga de crear un nuevo objeto Alojamiento.

\* Inicializa los atributos capacidad, precioPorPersona y distancia con los

\* valores de las constantes. Inicializa el arreglo de estudiantes<br />

\*

\*/

/\*\*

\* Nombre del metodo: albergarEstudiante<br />

\* <br />

\* Descripcion: Este metodo se encarga de guardar en una posicion libre del

\* arreglo de estudiantes el estudiante recibido como

\* parametro<br />

\* <br />

\* <b>Pre: </b> el arreglo de estudiantes debe estar inicializado. <br />

\* <b>Post: </b> el arreglo de estudiantes debe quedar con una posicion mas

\* ocupada.

\*

\* **@param** Estudiante

\* - objeto de tipo estudiante que representa al nuevo huesped

\* del alojamiento. estudiante != null

\*

\* **@throws** Exception

\* - lanza una excepcion si no se puede asignar al estudiante.

\* Puede utilizarse el metodo calcularOcupacion como ayuda.

\*

\*/

/\*\*

\* Nombre del metodo: calcularOcupacion<br />

\* <br />

\* Descripcion: Este metodo se encarga de retornar un valor entero que

\* indica cuantos estudiantes hay registrados en el alojamiento<br />

\* <br />

\* <b>Pre: </b> el arreglo de estudiantes debe estar inicializado.

\*

\* **@return** int - valor que representa el numero de estudiantes albergados en

\* el alojamiento.

\*

\*/

/\*\*

\* Nombre del metodo: calcularPrecioTotal<br />

\* <br />

\* Descripcion: Este metodo se encarga de retornar el valor que se debe

\* pagar por el alojamiento. Para ello se tiene en cuenta solo los

\* estudiantes albergados y el precio por persona. Debe incluir un impuesto

\* sobre el valor total. Este impuesto esta en la contante

\* PORCENTAJE\_IMPUESTO. Si esa constante tiene como valor, 20.4, entonces el

\* porcentaje de impuesto es 20.4%<br />

\* <b>Pre: </b> el arreglo de estudiantes debe estar inicializado.

\*

\* **@return** double - valor que representa el precio a pagar por el

\* alojamiento de los estudiantes.

\*

\*/

/\*\*

\* Nombre del metodo: darIdentificador<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* identificador.<br />

\*

\* **@return** String que contiene el valor del atributo identificador

\*/

/\*\*

\* Nombre del metodo: darCapacidad<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* capacidad.<br />

\*

\* **@return** int que contiene el valor del atributo capacidad

\*/

/\*\*

\* Nombre del metodo: darPrecioPorPersona<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* precioPorPersona.<br />

\*

\* **@return** double que contiene el valor del atributo precioPorPersona

\*/

/\*\*

\* Nombre del metodo: darDistancia<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* distancia.<br />

\*

\* **@return** double que contiene el valor del atributo distancia

\*/

/\*\*

\* Nombre del metodo: darEstudiantes<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor de la relacion

\* estudiantes.<br />

\*

\* **@return** Estudiante[] que representa la relacion con los

\* estudiantes

\*/

**Clase Vehiculo**

/\*\*

\* Nombre del metodo: Vehiculo (constructor)<br />

\* <br />

\* Descripcion: este metodo se encarga de crear un nuevo objeto Vehiculo.

\* Igualmente, se encarga de hacer el llamado a los metodos definidos para

\* inicializar los atributos capacidadKilometrosPorGalon.<br />

\* <br />

\*

\* **@param** placa

\* - String que contiene la placa del vehiculo

\* **@param** tipoVehiculo

\* - String que indica el tipo del vehiculo

\* **@throws** Exception

\* - Lanza una excepcion si la placa no tiene el formato AAA###

\* (tres vocales en mayúscula seguido de tres numeros enteros)

\* **@throws** Exception

\* - Lanza una excepción si el tipo de vehiculo NO es una cadena

\* con alguno de los valores definidos en las constantes

\*/

/\*\*

\* Nombre del metodo: asignarCapacidades<br />

\* <br />

\* Descripcion: Este metodo se encarga de inicializar el atributo que hace

\* referencia a la capacidad de kilometros por galon del vehiculo. Todo

\* depende del tipo de vehiculo: si es un vehiculo tipo economico tiene un

\* rendimiento de 60 kilometros por galon; si es un vehiculo tipo medio

\* tiene un rendimiento de 45 kilometros por galon; si es un vehiculo tipo

\* lujo, tiene un rendimiento de 30 kilometros por galon.<br />

\* <br />

\* <b>Pre: </b> el atributo tipoVehiculo debe estar inicializado con uno de

\* los valores definidos en las constantes.

\*/

/\*\*

\* Nombre del metodo: calcularCantidadGasolinaNecesaria<br />

\* <br />

\* Descripcion: Este metodo determina la cantidad de galones de gasolina

\* necesaria para recorrer la distancia al alojamiento. Para ello se tiene

\* en cuenta la capacidad de kilometros por galon del vehiculo.<br />

\*

\* **@return** double que corresponde a la cantidad de galones necesaria para

\* recorrer la distancia hasta el alojamiento. Si no hay alojamiento

\* asignado, entonces la cantidad a retornar es 0.

\*/

/\*\*

\* Nombre del metodo: esVocal (opcional)<br />

\* <br />

\* Descripcion: este metodo opcional permite determinar si un caracter es

\* una vocal o no.<br />

\* <br />

\*

\* **@param** letra

\* - char que corresponde al caracter que se desea evaluar

\* **@return** boolean que indica si es vocal (true) o si no lo es (false)

\*/

/\*\*

\* Nombre del metodo: cambiarAlojamiento<br />

\* <br />

\* Descripcion: este metodo se encarga de asignar el parametro a la

\* relacion alojamiento.<br />

\*

\* **@param** Alojamiento

\* - objeto que representa la relacion que va a tener el vehiculo

\* con un alojamiento<br />

\* <br />

\* <b>Post:</b> la relacion alojamiento debe cambiar por el

\* parametro recibido

\*/

/\*\*

\* Nombre del metodo: darPlaca<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* placa.<br />

\*

\* **@return** String que contiene el valor del atributo placa

\*/

/\*\*

\* Nombre del metodo: darTipoVehiculo<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* tipoVehiculo.<br />

\*

\* **@return** String que contiene el valor del atributo tipoVehiculo

\*/

/\*\*

\* Nombre del metodo: darCapacidadKilometrosPorGalon<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor del atributo

\* capacidadKilometrosPorGalon.<br />

\*

\* **@return** double que contiene el valor del atributo

\* capacidadKilometrosPorGalon

\*/

/\*\*

\* Nombre del metodo: darAlojamiento<br />

\* <br />

\* Descripcion: este metodo se encarga de retornar el valor de la relacion

\* alojamiento.<br />

\*

\* **@return** Alojamiento que representa la relacion con el alojamiento

\*/