Reading by abstraction: Agenda

Constructor

Línea 26: Constructor de la clase agenda. La inicializa como vacía, poniendo a 0 el número de entradas.

Línea 22-29: Es el constructor de la clase agenda.

addEntry

Línea 47: La variable booleana *found* se actualiza a true.
Línea 50: Pasamos a analizar la siguiente entrada de la variable de tipo AgendaNode *cur.*
Línea 46: Si el atributo nombre en la entrada de la variable *cur* coincide con el que le pasamos por parámetro, entraremos en el if.
Línea 45: Si *cur* no está vacío y la variable de control de tipo boolean sigue siendo false, nos mantendremos en el bucle.
Línea 44: Asignamos *cur* en la primera posición.
Línea 43: Si la agenda no está vacía, entramos en el if.

Línea 55: Asignamos en *first* una nueva entrada del tipo AgendaNode y retornamos true.
Línea 54: Entraremos en el if, si la variable *found* es false.
Línea 57: Si por el contrario *found* es true, retornaremos false.

Línea 40: Declaramos una variable de tipo AgendaNode llamada *cur*.
Línea 41: Declaramos una variable de tipo booleano llamada *found*, la cual se inicializa a false.

Líneas 39-59: El método addEntry que recibe por parámetro una variable de tipo Entry, añade un nuevo contacto en la agenda siempre que, previamente, dicho contacto no estuviera ya en ella. Si la agenda estuviera vacía, dicho contacto se colocaría el primero de la lista.

removeEntry

Línea 69: Si la variable *first* es null se retorna false.

Línea 73: Se crea una nueva variable de tipo AgendaNode y se inicializa a *first*.
Línea 75: Pasamos a analizar la siguiente entrada de la variable de tipo AgendaNode *prev.*
Línea 74: Mientras el nodo siguiente no sea nulo y el nombre del nodo siguiente no coincide con el nombre buscado, permanecerá en el bucle.

Línea 78: Si el nodo siguiente es nulo retorna false.
Línea 81: Establece el nodo siguiente a su siguiente.
Línea 82: Disminuye en 1 el número de entradas.

Línea 68-85: Esta función eliminará un contacto de la agenda. Si estuviera vacía no se producirá ningún efecto. Analizaremos todos los contactos de la agenda hasta que encontremos el que deseemos borrar. Si lo encontramos, a parte de eliminarlo, se disminuye el número de entradas de la agenda; y si no, no se producirá ningún efecto.

removeFirst

Línea 94: Crea una variable *p* de tipo Entry y la inicializa a null.

Línea 97: Establece el valor de *p* al info de *first*.
Línea 98: Pasamos a analizar la siguiente entrada de la variable de tipo AgendaNode *first*
Línea 99: Disminuye en 1 el número de entradas.
Línea 96: Si *first* es distinto de null, entrará en el if

Línea 102: Devuelve *p*.

Líneas 93-103: El método removeFirst eliminará el primer contacto de la agenda. Si esta estuviera vacía, no se producirá ningún efecto, y si hubiera solamente un único elemento el resultado de la lista debería ser el vacío. Si el borrado tiene éxito, se disminuirá el número de entradas en 1.

saveAgenda

Línea 134: Si la variable success es false, la pone a true.
Línea 137: Inserta la información del nodo actual en la variable parser.
Línea 138: Actualiza el valor de *line*.
Linea 139: Imprime el valor de *line* en el fichero.

Línea 124: Se crea la variable cur de tipo AgendaNode y se inicializa a first.
Línea 125: Se crea la variable line de tipo String.
Línea 126: Se crea la variable success de tipo boolean y se inicializa a false.
Línea 127: Se crea la variable p de tipo Parser y se inicializa.
Línea 129: Se crea el fichero agendofile.txt.
Línea 130: Se crea la variable bufferescritura para el archivo agendofile.txt.
Línea 131: Escribe en el fichero agendofile.txt lo que está guardado en bufferescritura.
Línea 133: Mientras el nodo actual no sea null lo recorre guarda la información del nodo en el fichero.
Línea 142: Retorna el valor de success.

Línea 123-143: La función saveAgenda debe almacenar la agenda en el archivo de texto agendaFile.txt.

loadAgenda

Línea 160: Cierra el buffererin.
Línea 161: Retorna false.
Línea 159: Comprueba si la cadena está vacía.

Línea 169: Añade la entrada a auxEntry.
Línea 168: Se comprueba si la línea tiene datos.
Línea 171: Termina el do while con la condición de que la cadena mientras no sea nula siga ejecutándose.

Línea 154: Abre el archivo el archivo agenda.
Línea 155: Lo introduce en una variable tipo bufferedReader.
Línea 172: Cierra el fichero.
Línea 174: Retorna true.

Línea 153-176: La función loadAgenda debe cargar el la agenda del archivo agendaFile.txt, añadiendo todas las entradas al archivo.