

APELLIDOS _____ NOMBRE _____

DNI _____ ORDENADOR _____ GRUPO (A/B/C) _____

La tabla periódica de elementos químicos se organiza en siete filas que llevan el nombre de *periodos* y en dieciocho columnas que reciben el nombre de *grupos* o *familias*. Cada elemento pertenece a un grupo y a un periodo determinado.

Para realizar el examen definiremos el tipo de datos TTabla, un array unidimensional de listas enlazadas (tipo TLista), donde cada lista tendrá los elementos de un grupo ordenados por su número atómico.

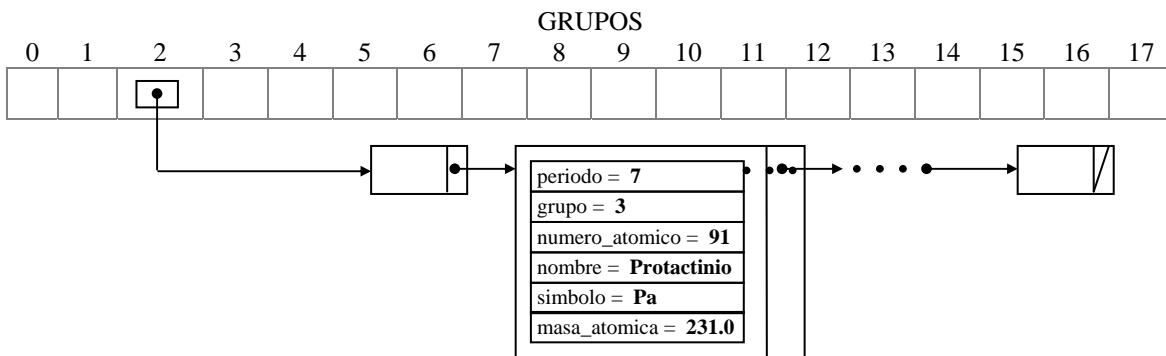


Fig. 1 Estructura de datos que usaremos para almacenar los elementos químicos de la tabla periódica

Cada elemento viene definido por un conjunto de propiedades entre las que consideraremos: *grupo* y *periodo* (ambos números naturales) al que pertenecen, *número atómico del elemento* (un número natural), *nombre*, *símbolo* (ambos cadenas de caracteres), y *masa atómica relativa* (número real).

El objetivo es diseñar e implementar un programa que permita manejar la tabla periódica a través del siguiente menú que contiene las **OPCIONES BÁSICAS** del programa:

```

Elaborado por: <Nombre> <Apellidos>          Curso 1º X Sistemas
Fecha: 25/06/2009                             Ordenador: <Ordenador>

          MENU EXAMEN JUNIO 2009
          =====
A. Leer las propiedades de un conjunto de elementos desde un fichero de
  texto e insertarlas en la tabla.
B. Mostrar el contenido de la tabla.
C. Buscar un elemento por su número atómico.
D. Eliminar un elemento de la tabla dado su número.

X. Salir del Programa
  
```

El formato del fichero de texto usado para almacenar los datos será:

```

<nombre><ENTER>
<periodo><ESP><grupo><ESP><símbolo><ESP><numero><ESP><masa><ENTER>
  
```

donde:

```

<periodo>  periodo al que pertenece el elemento
<grupo>    grupo al que pertenece el elemento
  
```

<nombre>	nombre del elemento
<símbolo>	símbolo que representa al elemento
<numero>	número atómico del elemento.
<masa>	masa atómica del elemento
<ESP>	espacio en blanco
<ENTER>	retorno de carro

El programa se va a dividir en los módulos **MElemento**, **MLista**, **MTabla** y **MErrores**. LAS OPERACIONES DESCRITAS A CONTINUACIÓN SON EL CONJUNTO MÍNIMO NECESARIO PARA IMPLEMENTAR LAS **OPCIONES BÁSICAS** DEL MENÚ PRINCIPAL:

- Gestión de datos de elementos químicos (**MElemento.h** y **MElemento.cpp**). Define el tipo **TElemento** y las funciones:
 - **escribirElemento(elemento)**: para **escribir los** datos de un elemento por pantalla.
 - **leerElementoFichero(fich, elemento)**: lee los datos de un elemento desde un fichero de texto, donde *fich* es el descriptor del mismo.

- Gestión de una lista enlazada con los datos de los elementos químicos (**MLista.h** y **MLista.cpp**). Define el tipo **TLista**, las constantes **ESP** (espacio) y **ENTER** y tendrá las siguientes operaciones:
 - **listaVacía (lista)**: devuelve cierto si la lista está vacía.
 - **crearLista (lista)**: crea una lista vacía.
 - **destruirLista (lista)**: destruye una lista previamente creada.
 - **insertarLista (lista, elemento, error)**: inserta en la lista un elemento ordenado por su número. Si el elemento ya existe se produce un error.
 - **eliminarLista (lista, numero, error)**: elimina de la lista un elemento dado su número. Si el elemento no existe se produce un error.
 - **buscarLista (lista, elemento, numero_atómico, error)**: devuelve, si existe, un elemento dado su número. En caso de que no exista se produciría un error.
 - **escribirLista (lista)**: imprime por pantalla la lista enlazada.

- Gestión de los datos de la tabla periódica (**MTabla.h** y **MTabla.cpp**). Define el tipo **TTabla** y todas las operaciones para su manejo:
 - **crearTabla (tabla)**: crea una tabla periódica vacía.
 - **insertarElemento (tabla, elemento, error)**: inserta un elemento en la posición de la tabla dada por su grupo. Si el elemento ya existe se produce un error.
 - **eliminarElemento (tabla, numero_atómico, error)**: elimina un elemento de la tabla dado su número. Si el elemento no existe se produce un error.
 - **buscarElemento (tabla, elemento, numero_atómico, error)**: devuelve, si existe, un elemento dado su número. En caso de que no exista se produciría un error.
 - **mostrarTabla (tabla)**: muestra los elementos de la tabla.
 - **destruirTabla (tabla)**: elimina todos los elementos de la tabla.
 - **cargarTablaFichero (nombre_fichero, tabla, error)**: inserta en una tabla los datos de los elementos guardados en un fichero de texto con el formato especificado en el enunciado.

- Módulo para la gestión de errores (**MErrores.h** y **MErrores.cpp**). Definirá el tipo enumerado **TError** con todos los errores posibles en la aplicación y el procedimiento **MuestraError (error)** que mostrará por pantalla el error ocurrido.

Los **ALUMNOS QUE NO TENGAN SUPERADO EL TRABAJO EN CLASE** (tienen una **nota** acumulada < 2), además tendrán que añadir al menú principal la siguiente opción:

E. Ordenar los elementos de un grupo por sus símbolos.

para ello tendrán que extender los módulos MTabla y MLista añadiéndole las siguientes operaciones:

▪ **MTabla:**

- **ordenaGrupo (tabla, grupo):** muestra por pantalla los símbolos de los elementos de un grupo ordenados por orden alfabético.

▪ **MLista**

- **ordenarLista (lista):** muestra por pantalla los símbolos de una lista de elementos ordenados por orden alfabético.

Una vez completadas las operaciones básicas, **PARA SUBIR NOTA** proceda a implementar las siguientes **OPCIONES ADICIONALES:**

F. Filtrar la tabla. El programa pedirá dos valores reales al usuario (masa mínima y masa máxima) y mostrará todos los elementos de la tabla cuya masa atómica esté entre el mínimo y el máximo (ambos incluidos).

G. Salvar los datos de la tabla en un fichero binario.

H. Cargar los datos de un fichero binario a una tabla.

Para implementar las opciones adicionales será necesario extender los módulos MTabla, MLista y MElemento añadiéndole las siguientes operaciones:

▪ **MTabla:**

- **filtrarTabla (tabla, min, max):** muestra por pantalla los elementos de la tabla cuya masa atómica esté entre min y max (ambos incluidos).
- **cargarFicheroBIN (nombre_fichero, tabla, error):** lee el contenido del fichero con nombre nombre_fichero y lo almacena en una tabla.
- **salvarFicheroBIN (nombre_fichero, tabla, error):** vuelca al fichero con nombre nombre_fichero el contenido de una tabla.

▪ **MLista**

- **filtrarLista (lista, min, max):** dada una lista de elementos muestra por pantalla aquellos cuya masa atómica esté entre min y max (ambos incluidos).
- **escribirFicheroBIN (fich, lista, error):** vuelca al fichero con nombre nombre_fichero el contenido de una tabla.

▪ **MElemento**

- **leerElementoFicheroBIN (fich, elemento, error):** lee el contenido del fichero con nombre nombre_fichero y lo almacena en una tabla.
- **escribirElementoFicheroBIN (fich, elemento, error):** vuelca al fichero con nombre nombre_fichero el contenido de una tabla.

NOTAS:

- **SE PODRÁN IMPLEMENTAR TODOS LOS PROCEDIMIENTOS Y FUNCIONES AUXILIARES QUE SE CONSIDEREN NECESARIOS DENTRO DEL PROGRAMA PRINCIPAL O EN LA PARTE DE IMPLEMENTACIÓN DE ALGUNO DE LOS MÓDULOS**
- **NO SE PODRÁ:**
 - AÑADIR OPERACIONES EN LA PARTE DE DEFINICIÓN DE UN MÓDULO
 - EL USO DE DETALLES DE IMPLEMENTACIÓN DE LAS LISTAS DENTRO DEL PROGRAMA PRINCIPAL es decir, no se podrá hacer nada del tipo `tabla=NULL`, `tabla-> sig =...` etc., fuera de la parte de implementación de los módulos de listas.
 - Suponer que los nombres no llevan espacios.
- **SE DEBE HACER UNA PAUSA DESPUÉS DE EJECUTAR CADA OPCIÓN DE MENÚ, Y BORRAR LA PANTALLA TRAS LA PAUSA.**
- **RECORDAD QUE TODO FICHERO DE CABECERA DEBE LLEVAR SU `#ifndef #endif`**

Ejemplo: Fichero: `cabecera.h`

```
#ifndef _Cabecera_h_
#define _Cabecera_h_
// Contenido del archivo de Cabecera
#endif
```

- **ES OBLIGATORIO TRABAJAR EN EL DIRECTORIO C:\LPSJUN09.**
- **MÍNIMO OBLIGATORIO PARA APROBAR** para quien tenga **SUPERADO EL TRABAJO EN CLASE** (2 puntos):
 - correcta la definición de tipos y la modularización
 - funcionar **CORRECTAMENTE** las opciones señaladas en la parte básica (A, B, C, D, X).
- **MÍNIMO ADICIONAL OBLIGATORIO PARA APROBAR** para quien **NO** tenga **SUPERADO EL TRABAJO EN CLASE** (2 puntos):
 - funcionar **CORRECTAMENTE** todas las opciones señaladas en la parte básica (A, B, C, D, X), más la opción E.