

Programación I

Práctico de Máquina Integrador

Fecha de control: 13/11/2017. Entrega del archivo lista.h y programador.h por mail

Fecha de entrega: 22/11/2017

Grupo máximo 2 (dos)

Se desea realizar un sistema que administre un registro de programadores. Para esto se ha decidido mantener la información de los mismos utilizando una lista dinámica. Además de las funcionalidades básicas (ABM) de programadores se desean realizar búsquedas y almacenamiento en archivos.

Se pide:

1- Definir el TDA programador en el archivo **programador.h**, el mismo debe contener:

a- la definición del tipo **prog** que almacena los siguientes datos:

- Código prog (código identificador del programador, único, tipo seleccionado por el alumno)
- Nombre y apellido (string de 50 caracteres)
- Fecha de nac (como la trabajada en la materia)
- Email (string)
- Nivel de programador (existen 3 niveles: inicial, medio, avanzado, seleccionar tipo y forma de representación)
- Lenguajes que conoce (es un arreglo de enteros cuyos valores son 0 o 1, el 0 indica que no conoce el lenguaje, el 1 que conoce el lenguaje)

Nota: Los lenguajes son 7 siguiendo en orden de posición del arreglo;

Pos 0-----C

Pos 1-----JAVA

Pos 2-----PYTHON

Pos 3-----Haskell

Pos 4-----Prolog

Pos 5-----JavaScript

Pos 6---- PHP

b- Las operaciones para manipular el TDA programador

init: recibe como parámetros los valores de un programador y retorna un **prog**

mostrar: retorna los valores del programador de a un campo por vez. Es decir mostrar_cod, mostrar_nya, mostrar_fnac, mostrar_email, mostrar_nivel, mostrar_leng

modificar: los campos que debe permitir modificar son sólo: email y lenguajes; para esto debe realizar la operaciones mod_email y mod_leng.

2- Definir el TDA Lista dinámica en el archivo lista.h el cual almacena TDA programadores (definido previamente)

a- Definir la/s estructuras necesarias para implementar una Lista de programador

b. Definir las operaciones necesarias para implementar la lista

c- No definir funciones que USEN la lista (por ej. Buscar, cargar n), estas se definen en el archivo **mi_grupo.c**

Carreras: Ing. en Computación, Ing. en Informática, Lic. en Cs. de la Computación

3. realizar el programa **mi_grupo.c** el cual hace uso de ambas librerías definidas (programador.h y lista.h)

a- El programa debe realizar las siguientes funcionalidades:

- a) Cargar n programadores, solicitando la cantidad por pantalla. Utilice para esto una función carga programador que pide los datos y utiliza el init del TDA programador para luego almacenarlo en la lista.
- c) Buscar un programador en la lista de programadores por código de programador.
- d) Mostrar los datos de un programador (buscado en la lista previamente) por pantalla.
- e) Mostrar todos los datos de los programadores almacenados de un nivel solicitado (**realizar de forma recursiva**)
- f) Mostrar todos los datos de los programadores que conocen un lenguaje determinado ingresado por el usuario.
- g) Eliminar un programador de la lista. Para esto debe ingresar el código del programador.
- h) Cargar nuevos lenguajes a un programador por código del programador.
- i) Modificar el mail a un programador por código del programador.
- j) Hacer una copia de la lista en el archivo backup.txt de manera tal que almacene todos los datos de la lista queden legibles y almacenados en el archivo. **Nota:** Ver COMO guardar los datos codificados como por ej en el caso de los lenguajes que conoce el programador, no debe almacenar 0's y 1's, sino los lenguajes que conoce.

b- Realice un menú de opciones que permita realizar las funcionalidades solicitadas, en cualquier orden y tantas veces como el usuario lo desee. Considere que se realiza un programa que luego será utilizado por un usuario, es por esto **NO OLVIDAR los carteles!!**

Nota. Recuerde que en todos los casos debe manejar los TDA 's con las funciones propias del mismo sin violar la semántica correspondiente a la estructura de almacenamiento. Considere el uso de tipo y estructuras auxiliares en caso de ser necesarias.

Aclaración: mi_grupo.c debe ser nombrado con los apellidos de los alumnos del grupo, por ejemplo perez_lopez.c