Prueba Parcial I – Estructura de Datos II

Docente: Ing. Juan Carlos Zepeda, sección: 479

Haciendo uso del API de lectura/escritura sobre un archivo binario (DataFile), implementar las siguientes clases, que permitan crear estructuras de campos de tamaño fijo, con indicador de longitud y delimitados por un carácter.

1. Crear clase FixedSize_Register con atributos:
   - ✓ DataFile * file: Crear un archivo binario con estructura de registro. Asignar nombre.
   - ✓ int code
   - ✓ char * name: Reservar tamaño de 30 espacios
   - ✓ double salary
   - ✓ char * job: Reservar tamaño de 20 espacios
2. Crear clase KnownVarSize_Register con atributos:
   - ✓ DataFile * file: Crear un archivo binario con estructura de registro. Asignar nombre.
   - ✓ int code
   - ✓ int sizeName: Indica la cantidad de caracteres que hay que leer para el nombre
   - ✓ char * name
   - ✓ double salary
   - ✓ int sizeJob: indica la cantidad de caracteres que hay que leer para el tipo de trabajo
   - ✓ char * job
3. Crear clase DelimiterVarSize_Register con atributos:
   - ✓ DataFile * file: Crear un archivo binario con estructura de registro. Asignar nombre.
   - ✓ int code
   - ✓ char * name
   - ✓ double salary
   - ✓ char * job

   La separación de cada campo y registro dentro del archivo binario se dara mediante el carácter punto y coma ";"

Cada una de las clases contara con las siguientes funciones públicas:

a) void print_register( ): Imprime en consola, los campos del registro actual
b) char * toChar( ): Construye un buffer de datos con la estructura en bytes de campos del registro actual
c) void fromChar(char *): Inserta en cada campo, los valores a partir del buffer de datos enviado por parametro
d) void open_file (char *): Abre el archive con la estructura de registro
e) void write_into_file( ): Escribe al final del archivo binario, la estructura de datos actual
f) void read_from_file(int pos): Lee a partir de una posición (índice) un registro dentro del archivo binario
g) void close_file( ): Cierra el archivo con la estructura de registro
h) int get_size( ): Obtiene el tamaño total en bytes de la estructura de registro

4. Crear función main (principal) que permita el uso de los tres (3) tipos de estructuras de archivos.

Subir archivo comprimido ZIP, con proyecto desarrollado en C++ en el enlace "Prueba Parcial I" en la semana 3 de la clase en el portal.