Prueba de examinación (Periodo extraordinario) correspondiente a la unidad de aprendizaje:

Programación Avanzada

Considere el proyecto prog26.cbp, que se puede encontrar en el enlace:

https://github.com/progavan/2021-2/tree/main/U1\_Programacion\_orientada\_a\_objetos/prog26\_PD\_n\_DE\_ADTs/

En ese proyecto se utiliza (entre otras) la clase Phone\_Directory:

class Phone\_Directory {

public:

// …

/\*\* Cargar el archivo de datos que contiene el

\* directorio, establecer una conexi\'on con

\* la fuente de datos.

\* @param source\_name El nombre del archivo

\* (fuente de datos) con las entradas/elementos del

\* directorio telef\'onico.

\*/

void load\_data(const std::string& source\_name);

// …

/\*\* Escribir el contenido del directorio en el

\* archivo de datos.

\*/

void save();

// …

}; /\* end class Phone\_Directory \*/

En la versión presentada en clase durante el curso, los datos correspondientes a las entradas del directorio telefónico se encontraban en un archivo de texto llamado directorio.txt (que se puede encontrar también en la url de ubicación del proyecto indicada arriba). Debido a que el archivo directorio.txt es un archivo de caracteres, la resolución del método load\_data() utiliza la función getline(), como se muestra a continuación:

void Phone\_Directory::load\_data(const std::string& source\_name)

{

// Recordar el nombre de la fuente de datos

// para utilizarlo al salvar

this->source\_name = source\_name;

// Crear una cadena de entrada para este archivo.

ifstream in(source\_name.c\_str());

if(in){// Si existe el flujo

string name;

string number;

while(getline(in,name)){

if(getline(in,number)){

add(name,number);

}

}

// Cerrar el archivo

in.close();

}

}

Por otra parte, la resolución del método save(), utiliza un objeto de la clase ofstream, como se muestra a continuación:

void Phone\_Directory::save(){

if(modified){ // If not modified, do nothing

//Crear una cadena de salida.

ofstream out(source\_name.c\_str());

for(int i=0;i<size; i++){

out<<the\_directory[i].get\_name()<<std::endl;

out<<the\_directory[i].get\_number()<<std::endl;

}

//Cerrar la cadena de salida.

out.close();

modified=false;

}

}

Como examen extraordinario, modifique usted las resoluciones de los métodos load\_data() y save() de la clase Phone\_Directory, de tal manera que los datos del directorio se obtengan y se guarden desde y hacia un archivo binario llamado directorio.raw.

PISTAS:

Revise el objetivo 01\_AbsToExtFunc del proyecto prog33-Abstraccion.cbp, en el cual se leen datos del archivo binario contactos.raw. El proyecto prog33-Abstraccion.cbp se puede obtener en el enlace

https://github.com/progavan/2021-2/blob/main/U1\_Programacion\_orientada\_a\_objetos/prog33-Abstraccion/

En particular, revise el método ReadEntry() en el archivo FileMailingListEntry.h,

en este método se utiliza la función fread(), de la biblioteca estándar de C, para leer registros del archivo binario contactos.raw.

Revise también el archivo

https://github.com/progavan/2021-2/blob/main/U1\_Programacion\_orientada\_a\_objetos/prog33-Abstraccion/01\_AbsToExtFunc/02\_ch\_abstraction/02\_ch\_aux.cpp

en el cual se muestra como escribir en un archivo binario (contactos.raw) utilizando la función fwrite(), de la biblioteca estándar de C.

NOTA: TIENE VIERNES 7 DE ENERO DE 2022 A LAS 17:00, COMO FECHA Y HORA LIMITE PARA SUBIR A LA PLATAFORMA MS TEAMS EN LA SECCION DE TAREAS DEL EQUIPO

2022\_1\_PROG\_AVAN\_\_\_2MV10, EN EL CUAL USTED ESTA INSCRITO, UN ARCHIVO TXT CON UNICAMENTE LOS CODIGOS DE LOS METODOS load\_data() y save() (MODIFICADOS COMO SE SOLICITA) DE LA CLASE Phone\_Directory.