///////////////////////////////////////// PERSONA //////////////////////////////////////////////////

#ifndef PERSONA\_H

#define PERSONA\_H

#include <fstream>

#include <string>

#include <sstream>

using namespace std;

class Persona {

private:

int edad;

int id;

char genero;

string nombre;

public:

Persona() {

this->id = 0;

this->nombre = "Sin def";

this->edad = 0;

this->genero = 'x';

}

//----------------------------------------------------------------------------------------------

Persona(int auxID, string auxNombre, int auxEdad, char genero) {

this->id = auxID;

this->nombre = auxNombre;

this->edad = auxEdad;

this->genero = genero;

}

//----------------------------------------------------------------------------------------------

int getEdad() { return this->edad; }

int getId() { return this->id; }

void setNombre(string auxNombre) { this->nombre = auxNombre; }

string getNombre() { return this->nombre; }

void setEdad(int auxEdad) { this->edad = auxEdad; }

void setId(int auxId) { this->id = auxId; }

void setGenero(int auxgenero) { this->genero = auxgenero; }

int getGenero() { return this->genero; }

//----------------------------------------------------------------------------------------------

void guardarPersona(ofstream& file) {

char auxNombre[20];

strcpy\_s(auxNombre, nombre.c\_str());

file.write((char\*)&id, sizeof(int));

file.write((char\*)&auxNombre, sizeof(auxNombre)); // evitemos usar string

file.write((char\*)&edad, sizeof(int));

file.write((char\*)&genero, sizeof(char));

}

//----------------------------------------------------------------------------------------------

void cargarPersona(ifstream& file) {

char auxNombre[20];

file.read((char\*)&id, sizeof(int));

file.read((char\*)&auxNombre, sizeof(auxNombre));

file.read((char\*)&edad, sizeof(int));

file.read((char\*)&genero, sizeof(char));

nombre = auxNombre;

}

//----------------------------------------------------------------------------------------------

string toString()

{

stringstream s;

s << "\t" << "-id: " << this->getId() << endl;

s << "\t" << "-nombre: " << this->getNombre() << endl;

s << "\t" << "-edad: " << this->getEdad() << endl;

s << "\t" << "-genero: " << this->getGenero() << endl;

return s.str();

}

};

#endif

///////////////////////////////////////// CONTENEDOR ///////////////////////////////////////////////

#ifndef CONTENEDOR\_H

#define CONTENEDOR\_H

#include "Persona.h"

#include <iostream>

using namespace std;

class Contenedor {

private:

Persona \*v; // vector dinamico con objetos estaticos

int tam;

int can;

public:

Contenedor(int t = 10) {

can = 0;

tam = t;

this->v = new Persona[t];

}

~Contenedor() {

if (tam>0) { delete[] this->v; }

can = 0;

tam = 0;

}

//----------------------------------------------------------------------------------------------

void setTam(int tam) { this->tam = tam; }

int getTam() { return this->tam; }

void setCan(int can) { this->can = can; }

int getCan() { return this->can; }

//----------------------------------------------------------------------------------------------

Persona getPersona(int pos) { return this->v[pos]; }

//----------------------------------------------------------------------------------------------

bool setPersona(int pos, Persona&aux) {

if (pos < this->getTam() && pos >= 0) {

this->v[pos].setEdad(aux.getEdad());

this->v[pos].setNombre(aux.getNombre());

this->v[pos].setId(aux.getId());

this->v[pos].setGenero(aux.getGenero());

can++;

return true;

}

else {

cout << "Indice fuera del rango adecuado\n";

return false;

}

}

//-----------------------------------------------------------------------------------------------------

bool guardarContenedor(string archivo) {

ofstream myFile(archivo.c\_str(), ios::out);//se abre un archivo para escritura

//Nota: c\_string(), covierte el string a const char \*

if (myFile.is\_open())//si el archivo esta abierto

{

myFile.write((char\*)(&(this->can)), sizeof(int));

//Se desea guardar la cantidad de elementos

for (int i = 0; i < this->can; i++) {

this->v[i].guardarPersona(myFile);

}

myFile.close();//se cierra el archivo

return true;

}

else {

cout << "El archivo no se pudo abrir"<<endl;

return false;

}

}

//----------------------------------------------------------------------------------------------

bool cargarContenedor(string archivo) {

ifstream myFile(archivo.c\_str(), ios::in);// archivo para lectura

if (myFile.is\_open())//si el archivo esta abierto

{

myFile.read((char\*)(&(this->can)), sizeof(int));//se lee la cantidad de registros guardados

for (int i = 0; i<this->can; i++) {

this->v[i].cargarPersona(myFile);//cargo la info del archivo en cada instancia de persona

}

myFile.close();//cierro el archivo

return true;

}

else {

cout << "El archivo no se pudo abrir"<<endl;

return false;

}

}

};

#endif

///////////////////////////////////////// INTERFAZ /////////////////////////////////////////////////

#include <iostream>

using namespace std;

#include "Contenedor.h"

class Interfaz {

public:

void mostrar(string texto) {

cout << endl << texto << endl;

}

void mostrar(Contenedor &cont) {

cout << endl << "\t" << " -------------------------------------------------------------------" << endl;

for (int i = 0; i<cont.getCan(); i++) {

cout << cont.getPersona(i).toString() << endl;

}

cout << endl << "\t" << " -------------------------------------------------------------------" << endl;

}

};

///////////////////////////////////////// MAIN /////////////////////////////////////////////////

#include <iostream>

#include "Contenedor.h"

#include "interfaz.h"

#include "Persona.h"

void main() {

Contenedor cont1(10);

Contenedor cont2(10);

Interfaz\* inter = new Interfaz();

Persona p1(101, "Maria", 23, 'f');

Persona p2(102, "Juan", 32, 'm');

Persona p3(103, "Lucia", 56, 'f');

Persona p4(104, "Rosa", 18, 'f');

Persona p5(105, "Marco", 25, 'm');

cont1.setPersona(0, p1);

cont1.setPersona(1, p2);

cont1.setPersona(2, p3);

cont1.setPersona(3, p4);

cont1.setPersona(4, p5);

cont1.guardarContenedor("datoe.txt");

inter->mostrar((string)"Datos del contenedor1 que queremos guardar en el archivo");

// sin el casting da error

inter->mostrar(cont1);

cont2.cargarContenedor("datoe.txt"); // caRgamos con lo que esta guardado en el archivo

inter->mostrar((string)"Datos leidos en el archivo y guardados en el contenedor 2");

inter->mostrar(cont2);

cin.get();

cin.get();

}