

TDAs Lista.

Rodríguez Tabares Juan

Ingeniería en computación
Centro Universitario de Ciencias Exactas e Ingenierías
Universidad de Guadalajara

Abstract

Este trabajo comprende a la elaboración de la actividad numero 4 de la asignatura Estructura de datos II, de la carrera de Ingeniería en computación llevada acabo en el ciclo escolar 2020B.

1. Introducción

En este documento veremos como se adapto el programa entregado por el profesor, realizando los cambios necesarios para que funcione de forma correcta y optima.

Los cambios a realizar seran:

Añadir/Eliminar lo necesario para que el programa pueda importar datos de personas desde un archivo existente.

2. Desarrollo

En este punto veremos como se adapto el programa a los requerimientos pedidos paso a paso para llevar acabo su correcto funcionamiento.

NOTA: Se omitio la adición de el codigo Busquedainterna.h ya que no se realizo ninguna modificacion a este.

2.1. Arreglos.h

Las modificaciones hechas en esta parte en resumen se concentran en la adición de dos funciones y la eliminación de una, la explicación de estas estará a continuación:

Línea 16:

Adición de la función *agregar* que tiene como parametros un objeto del tipo T, Un arreglo del tipo T, y una variable del tipo int.

Línea 28-32:

El funcionamiento de esta consiste en añadir en el arreglo (segundo parametro) el tipo de dato T (primer parametro) en la posición pasada (tercer parametro). Al final de la función esta asigna el valor de i al tamaño (en incremento).

Línea 19: Adición de la función *ReadFromDisk* la cual recibe como parametro un arreglo del tipo T y a su vez devuelve un arreglo del mismo tipo.

Línea 59-67: El funcionamiento de la función *ReadFromDisk* consiste en declarar un arreglo llamado *ara*, pasando ha asignar los valores de la función leer a el arreglo *ara* y asignando los valores de esta a nuestro parametro *arr* retornando este arreglo anteriormente mencionado.

Nota: Fue eliminada la función *Lectura* ya que con los cambios adoptados la función quedo totalmente obsoleta

2.2. Persona.h

Las modificaciones de esta parte del trabajo solo consiste en la adición de una función la cual se explicara a continuación:

2.3 main.cpp

Línea 20: Adición de la función Leer que tiene como parametro un arreglo del tipo Persona y a su vez devuelve un arreglo del mismo tipo.

Línea 61-91: Definición de la función Leer se añadieron 2 nuevas variables del tipo int: *j e i*, se añadió el bucle while para recorrer el archivo y mientras esto se realiza poder leer los datos guardandolos en el objeto declarado *per* del tipo Persona y en el arreglo pasado como parametro agregar este objeto *per* en la posición *j* del arreglo.

Si la lectura llega a su fin se cierra el archivo y se retorna el arreglo modificado.

2.3. main.cpp

Esta parte del trabajo recibió un gran cambio ya que se modificó una gran parte para poder adaptar el programa a las funciones anteriormente realizadas, a continuación la explicación de las modificaciones:

Línea 11: declaración de *file* que contendrá al archivo para llevar a cabo la importación.

Línea 16: Declaración de una nueva variable del tipo int, llamada *aux*.

Línea 18-20: Comprobación para saber si el archivo existe, si este existe se agregan los datos dentro por las funciones anteriormente explicadas, si no existe sigue el programa sin importar datos.

Línea 21-52: Se añadió un do-while para poder realizar un menú y hacer el programa un poco más amigable para el usuario.

Línea 24-51: Se añadió una función switch para llevar a cabo la ejecución correcta del menú, a continuación la explicación de cada uno de los casos.

Línea 25-28, caso 1: Se pide al usuario añadir los valores del objeto *per* que es del tipo Persona. Siguiendo se procede a añadir a la persona con el método agregar anteriormente explicado. Al final de este caso se aumenta *aux* en 1.

Línea 31-41, caso 2: Se realiza la búsqueda respetando el diseño original del main.cpp proporcionado por el profesor.

Línea 43-50, caso 3,4: Simples validaciones para salir/advertir.

Línea 54: Escritura en el archivo. Línea 55: cierre del archivo.

2.4. Código en C++

Arreglos.h: Aquí se pueden observar los cambios realizados que se explicaron en la parte 2.1 del desarrollo.

```
1 #ifndef ARREGLOS_H_INCLUDED
2 #define ARREGLOS_H_INCLUDED
3 #include<iostream>
4 #include <fstream>
5 #include <utility>
6 using namespace std;
7 #define MAX 100
8 template <class T>
9 class Arreglo
10 {
11 private:
12     T Datos[MAX];
13     int Tam;
14 public:
15     Arreglo();
16     void agregar(T&, Arreglo<T>&, int i);
17     void Escribe();
18     void EscribeFile();
19     Arreglo<T> ReadFromDisk(Arreglo<T>&);
20     int RegresaTam();
21     T RegresaValor(int);
22 };
23 template <class T>
24 Arreglo<T>::Arreglo()
25 {
26     Tam= 0;
27 }
28 template <class T>
29 void Arreglo<T>::agregar(T& per, Arreglo<T>& arr, int i){
30     arr.Datos[i] = per;
31     arr.Tam = i;
32 }
```

2.4 Código en C++

```
33
34
35 template <class T>
36 void Arreglo<T>::Escribe()
37 {
38     int Indice;
39     if (Tam > 0)
40     {
41         std::cout << "\n Impresin de datos\n";
42         for (Indice= 0; Indice < Tam; Indice++)
43             std::cout << "\t" << Datos[Indice];
44     }
45     else
46         std::cout << "\nNo hay elementos registrados.";
47 }
48
49 template <class T>
50 void Arreglo<T>::EscribeFile()
51 {
52     int Indice;
53     for (Indice= 0; Indice <= Tam; Indice++)
54     {
55         Datos[Indice].Capturar();
56     }
57 }
58
59 template <class T>
60 Arreglo<T> Arreglo<T>::ReadFromDisk(Arreglo<T>& arr)
61 {
62     Arreglo ara;
63     T dato, aux;
64     arr = dato.Leer(ara);
65     return arr;
66     std::cout << "Datos traídos del disco" << '\n';
67 }
68
69 template <class T>
70 int Arreglo<T>::RegresaTam()
71 {
72     return Tam;
73 }
74
75 template <class T>
76 T Arreglo<T>::RegresaValor(int Indice)
77 {
78     return Datos[Indice];
79 }
80
81 #endif // ARREGLOS_H_INCLUDE
```

Persona.h.cpp: Aquí se pueden observar los cambios realizados en la clase persona con mayor exactitud, recuerde que esta parte se explica en la parte 2.2 del desarrollo.

```
1 #ifndef PERSONA_H_INCLUDED
2 #define PERSONA_H_INCLUDED
3 #include <iostream>
4 #include <fstream>
5 #include <Cstring>
6 #include <stdlib.h>
7 #include <cstdlib>
8 #include <utility>
9 #include "Arreglos.h"
10 using namespace std;
11 class Persona
12 {
13 private:
14     int AnioNac;
15     char NomPers[64], LugNac[64];
16 public:
17     Persona();
18     Persona(int, char[], char[]);
```

2.4 Código en C++

```
19 void Capturar();
20 Arreglo<Persona> Leer(Arreglo<Persona>&);
21 int operator != (Persona);
22 int operator > (Persona);
23 int operator < (Persona);
24 friend std::istream &operator >> (std::istream &, Persona &);
25 friend std::ostream &operator << (std::ostream &, Persona &);
26 };
27
28
29 Persona::Persona()
30 {}
31
32 Persona::Persona(int ANac, char NomP[], char LugN[])
33 {
34     AnioNac= ANac;
35     strcpy(NomPers, NomP);
36     strcpy(LugNac, LugN);
37 }
38
39 void Persona::Capturar()
40 {
41     int a,b,c;
42     ofstream o("archivo.txt", ios::app);
43     if(!o.good()){
44         cout<<"error al abrir archivo"<<endl;
45     }
46     else{
47         a=strlen(NomPers);
48         b=strlen(LugNac);
49         c=strlen((char *)&AnioNac);
50         o.write((char *)&a, sizeof(int));
51         o.write((char *)&NomPers, a);
52         o.write((char *)&b, sizeof(int));
53         o.write((char *)&LugNac, b);
54         o.write((char *)&c, sizeof(int));
55         o.write((char *)&AnioNac, c);
56     }
57     o.close();
58 }
59
60
61 Arreglo<Persona> Persona::Leer(Arreglo<Persona>& arr)
62 {
63     int a, b, c,i,j=0;
64     ifstream o("archivo.txt");
65     if(!o.good()){
66         cout<<"error al abrir archivo"<<endl;
67     }
68     while(!o.eof()){
69         fflush(stdin);
70         a=strlen(NomPers);
71         b=strlen(LugNac);
72         c=strlen((char *)&AnioNac);
73         o.read((char *)&a, sizeof(int));
74         o.read((char *)&NomPers, a);
75         NomPers[a] = '\0';
76         o.read((char *)&b, sizeof(int));
77         o.read((char *)&LugNac, b);
78         LugNac[b] = '\0';
79         o.read((char *)&c, sizeof(int));
80         o.read((char *)&AnioNac, c);
81         i = AnioNac;
82         Persona per(i,NomPers,LugNac);
83         arr.agregar(per,arr,j);
84         j++;
85         if(o.eof()){
86             o.close();
87             return arr;
88         }
89     }
```

2.4 Código en C++

```
90  o.close();
91  }
92
93
94
95  int Persona::operator != (Persona Pers)
96  {
97      int Resp=0;
98      if (strcmp(NomPers, Pers.NomPers) != 0)
99          Resp= 1;
100     return Resp;
101 }
102
103
104  int Persona::operator > (Persona Pers)
105  {
106      int Resp=0;
107      if (strcmp(NomPers, Pers.NomPers) > 0)
108          Resp= 1;
109      return Resp;
110 }
111
112  int Persona::operator < (Persona Pers)
113  {
114      int Resp=0;
115      if (strcmp(NomPers, Pers.NomPers) < 0)
116          Resp= 1;
117      return Resp;
118 }
119
120  std::istream &operator >> (std::istream &Lee, Persona &ObjPers)
121  {
122      std::cout <<"\n\nIngrese nombre de la Persona: ";
123      Lee>> ObjPers.NomPers;
124      std::cout <<"\n\nIngrese a o de nacimiento: ";
125      Lee>> ObjPers.AnioNac;
126      std::cout <<"\n\nIngrese lugar de nacimiento: ";
127      Lee>> ObjPers.LugNac;
128      return Lee;
129  }
130  std::ostream &operator << (std::ostream &Escribe, Persona &ObjPers)
131  {
132      Escribe<<"\n\nDatos de la Persona\n";
133      Escribe<<"\nNombre: "<<ObjPers.NomPers;
134      Escribe<<"\nLugar de nacimiento: "<<ObjPers.LugNac;
135      Escribe<<"\n A o de nacimiento: "<<ObjPers.AnioNac;
136      return Escribe;
137  }
138
139
140  #endif // PERSONA_H_INCLUDED
```

main.cpp: Es donde se mostrara la funcion principal del programa recuerde que la explicacion de las modificaciones esta en la parte 2.3 del desarrollo.

```
1  #include <iostream>
2  #include <fstream>
3  #include "BusquedaInterna.h"
4  #include "Persona.h"
5  #include "Arreglos.h"
6
7  using namespace std;
8
9  int main()
10 {
11     ifstream file("archivo.txt");
12     // SecuencialDesord<Persona> Buscador;
13     Binaria<Persona> Buscador;
14     Persona per;
15     Arreglo<Persona> Asistentes;
16     int Resp,opc,tam,aux=0;
17     char Nom[64], nac[3]={" "};
```

2.4 Código en C++

```
18  if (file) {
19      Asistentes = Asistentes.ReadFromDisk(Asistentes);
20  }
21  do {
22      std::cout << "Ingrese opcion\n1) Aadir\n2) Buscar\n3) Salir" << '\n';
23      cin>>opc;
24      switch (opc) {
25          case 1:{
26              cin>>per;
27              Asistentes.agregar(per,Asistentes,aux);
28              aux++;
29          }break;
30
31          case 2:{
32              cout<<"\nIngresa el nombre de la persona a buscar: ";
33              cin>>Nom;
34              Persona Alguen(0,Nom,nac);
35              Resp= Buscador.Busca(Asistentes, Alguen);
36              if (Resp >= 0){
37                  Alguen=Asistentes.RegresaValor(Resp);
38                  cout<<Alguen;
39              }
40              else cout<<"\n\nNO se encontr a la persona\n\n";
41          }break;
42
43          case 3:{
44              std::cout << "Saliendo..." << '\n';
45              break;
46          }break;
47
48          default:{
49              std::cout << "Opcion invalida" << '\n';
50          }break;
51      }
52  } while (opc!=3);
53
54  Asistentes.EscribeFile();
55  file.close();
56  return 0;
57
58 }
```

3. Resultados

```
Ingrese opcion
1)Agregar
2)Buscar
3)Salir
1

Ingrese nombre de la Persona: Juan

Ingrese a o de nacimiento: 2000

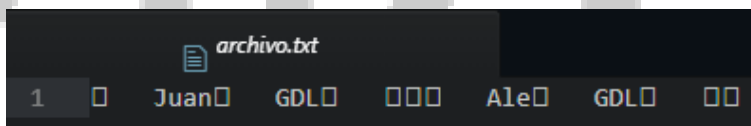
Ingrese lugar de nacimiento: GDL
Ingrese opcion
1)Agregar
2)Buscar
3)Salir
1

Ingrese nombre de la Persona: Ale

Ingrese a o de nacimiento: 2000

Ingrese lugar de nacimiento: GDL
```

Figura 1: Primera parte de la ejecución del programa la cual consiste en el registro de las personas.



id	nombre	lugar	fecha
1	Juan	GDL	2000
	Ale	GDL	2000

Figura 2: Vista de la generacion del archivo con el contenido que se le fue añadido en el programa.

```
Datos traídos del disco

Ingrese opción
1) Añadir
2) Buscar
3) Salir
2

Ingrese el nombre de la persona a buscar: Juan

Datos de la Persona
Nombre: Juan
Lugar de nacimiento: GDL
Año de nacimiento: 2000
Ingrese opción
1) Añadir
2) Buscar
3) Salir
2

Ingrese el nombre de la persona a buscar: Ale

Datos de la Persona
Nombre: Ale
Lugar de nacimiento: GDL
Año de nacimiento: 2000
Ingrese opción
1) Añadir
2) Buscar
3) Salir
```

Figura 3: Carga de datos del archivo con éxito!.

4. Conclusiones

La realización de este trabajo fue bastante entretenida y curiosa, ya que realice partes de código que nunca había realizado, pero el resultado me deja conmovido, queda aclarar que esta práctica en lo personal fue más fácil que en la anterior (Actividad 03).