

6-9-2021



Estructuras anidadas

Practica: 1

Materia: Estructura de datos

Sección: D01.

Código: 216584703

Carrera: Ingeniería en computación.

Nombre alumno: Padilla Pérez Jorge
Daray

Nombre profesor: Julio Esteban
Valdes Lopez

Índice general

Índice	1
Introducción	2
Pantallazos	3-6
Conclusión	7
Código fuente	8...

Introducción

Mi práctica consiste en la implementación de una anidación de estructuras anidadas, yo lo realice en lenguaje c++, en el cual aparte de implementar la anidación de las estructuras, se implementa con registros para tener un control de los datos ingresados por el usuario, en este caso se hizo una estructura de dirección, y otra de persona en la cual esta anidada la dirección de esta persona con un limite de registro de 10 personas.

En el mismo programa se incluyen funciones especificas que serian para insertar los elementos, para imprimir estos mismos, para buscar los datos por su posición, y una función para eliminar el nodo seleccionado.

Pantallazos

```
1 #include <iostream>
2 #include <cstring>
3 #include <cstdlib>
4 #include <string.h>
5
6 using namespace std;
7
8 const int MaxData=10;
9
10 struct Address{
11     char street[100];
12     char city[50];
13     char state[20];
14 };
15
16 struct persona{
17     char name[50];
18     struct Address Address_persona;
19     int codigo_postal;
20 }data[10];
21
22
23 int head=0;
24 int actual=0;
25 int ultimo=0;
```

En este pedazo de código se aprecia la estructura address con sus atributos, y debajo de esta la estructura persona la cual tiene sus atributos y la anidación de la estructura address, con un registro de 10 personas y se inicializa la cabeza en 0.

Aquí se aprecia la función insertar la cual sirve para insertar los elementos que se le piden al usuario, y se terminan guardando en el registro ultimo de este.

```
27 void inserta(){
28     ultimo++;
29     fflush(stdin);
30     cout<<"Give the name: ";
31     cin.getline(data[ultimo].name,50);
32     cout<<"Give the street ";
33     cin.getline(data[ultimo].Address_persona.street,100);
34     cout<<"Give the city ";
35     cin.getline(data[ultimo].Address_persona.city,50);
36     cout<<"Estado: ";
37     cin.getline(data[ultimo].Address_persona.state,20);
38     cout<<"codigo postal: ";
39     scanf("%i",&data[ultimo].codigo_postal);
40 }
```

```
42 void siguiente() {
43     cout<<"\nName: "<<data[actual].name<<endl;
44     cout<<"street: "<<data[actual].Address_persona.street<<endl;
45     cout<<"City: "<<data[actual].Address_persona.city<<endl;
46     cout<<"State: "<<data[actual].Address_persona.state<<endl;
47     cout<<"pin: "<<data[actual].codigo_postal<<endl;
48     actual++;
49
50     if (actual >MaxData || actual>ultimo){
51         cout<<"ha llegado al ultimo nodo"<<endl;
52         system("pause");
53         actual--;
54     }
55 }
56
57 void imprimeLista() {
58     int i;
59     actual=head;
60
61     for(i=head; i<= ultimo; i ++){
62         siguiente();
63     }
```

aquí se aprecia la función siguiente e imprimir, la cuales se encargan de que cada vez que imprime actual incrementa de valor y se comprueba si se ha llegado al ultimo nodo.

aquí la función buscar tiene de parámetro la posición de este, se evalúa primero si esa posición existe y si esta existe imprime los contenidos de dicha posición, si la posición no existe manda mensaje de no encontrada y retorna al menú principal.

```
64 void buscar(int pos) {
65     cout<<"posicion: "<<pos<<endl;
66     if (pos > ultimo || pos <head ) {
67         cout<<"posicion no encontrada"<<endl;
68         return;
69     }
70     cout<<"\nName: "<<data[pos].name<<endl;
71     cout<<"street: "<<data[pos].Address_persona.street<<endl;
72     cout<<"City: "<<data[pos].Address_persona.city<<endl;
73     cout<<"State: "<<data[pos].Address_persona.state<<endl;
74     cout<<"pin: "<<data[pos].codigo_postal<<endl;
75     pos++;
76 }
77
```

| Pantallazos

Aquí se implementa la función `remove` la cual primero pregunta que índice `remove` y se comprueba que este en el rango, después se hace un `for` el cual nos permite copiar los valores sumándoles uno al índice, y decrementando el valor de ultimo al ser removido, también `actual` se iguala a la cabeza.

```
78 void remove() {
79     int indice;
80     cout<<"Que nodo quieres remove?"<<endl; cin>>indice;
81     indice--;
82     if (indice > ultimo || indice < head) {
83         cout<<"indice fuera de rango"<<endl;
84         return;
85     }
86     int i;
87     for(i=indice; i<=ultimo; i++){
88         strcpy(data[i].name, data[i+1].name);
89         strcpy(data[i].Address_persona.city, data[i+1].Address_persona.city);
90         strcpy(data[i].Address_persona.street, data[i+1].Address_persona.street);
91         strcpy(data[i].Address_persona.state, data[i+1].Address_persona.state);
92         data[i].codigo_postal = data[i+1].codigo_postal;
93     }
94     if(indice == MaxData-1){
95         strcpy(data[indice].name, "");
96         strcpy(data[indice].Address_persona.city, "");
97         strcpy(data[indice].Address_persona.street, "");
98         strcpy(data[indice].Address_persona.state, "");
99         data[indice].codigo_postal=0;
100     }
101     ultimo--;
102     actual=head;
103 }
```

aquí se aprecia el menú.

```
105 int main()
106 {
107     int opcion, pos;
108     while(opcion!=0){
109         cout<<"Que opcion deseas, \n1-insertar \n2-imprimir lista \n3-siguiente nodo \n4-remove por nodo \n5-buscar por posicion \n6-salir \nopcion: "<<endl;
110         cin>>opcion;
111         switch(opcion){
112             case 1: system("cls");
113                     inserta();
114                     break;
115             case 2: system("cls");
116                     imprimirlista();
117                     break;
118             case 3: system("cls");
119                     siguiente();
120                     break;
121             case 4: system("cls");
122                     remove();
123                     system("pause");
124                     break;
125             case 5: system("cls");
126                     cout<<"Que posicion deseas consultar : "<<endl; cin>>pos;
127                     buscar(pos);
128                     break;
129             case 6: break;
130             default: printf("Esa opcion no existe...\n");
131         }
132     }
133     return 0;
134 }
```

| Pantallazos

```
Que opcion deseas,  
1-insertar  
2-imprime lista  
3-siguiente nodo  
4-remove por nodo  
5-buscar por posicion  
6-salir  
Opcion:
```

Este es el menú del usuario

```
Give the name: a  
Give the street a  
Give the city a  
Estado: a  
codigo postal: 1  
Que opcion deseas,  
1-insertar  
2-imprime lista  
3-siguiente nodo  
4-remove por nodo  
5-buscar por posicion  
6-salir  
Opcion:
```

Se insertan los elementos una vez hecho te dirige al menú.

En esta otra se imprime todo.

```
Name: a  
street: a  
City: a  
State: a  
pin: 1  
  
Name: b  
street: b  
City: b  
State: b  
pin: 2  
  
Name: c  
street: c  
City: c  
State: c  
pin: 3  
ha llegado al ultimo nodo  
Presione una tecla para continuar . . .
```

```
Name: b  
street: b  
City: b  
State: b  
pin: 2  
  
Name: c  
street: c  
City: c  
State: c  
pin: 3  
ha llegado al ultimo nodo  
Presione una tecla para continuar . . .
```

En este se elimina el primer nodo

```
Que posicion deseas consultar :  
1  
posicion: 1  
  
Name: c  
street: c  
City: c  
State: c  
pin: 3  
Que opcion deseas,  
1-insertar  
2-imprime lista  
3-siguiente nodo  
4-remove por nodo  
5-buscar por posicion  
6-salir  
Opcion:
```

Aquí por ultimo se aprecia que al buscar por posición se busca la posición 1 lo cual muestra el segundo nodo porque las posiciones en un arreglo son de 0 en adelante.

Conclusión

Respecto a la realización del código concluyo que tarde mas de lo esperado poder realizarlo ya que no contaba con la información correcta del mismo por haber faltado a la clase en la que se dejo la actividad, sin embargo, gracias a mis compañeros pude llegar a determinar como era el programa solicitado, y su realización.

Un aspecto negativo es que no pude hacer que al momento de llenarse el registro este se vaciara y comenzara de nuevo, intente muchas veces pero no se logro el cometido, espero que en la clase se pueda solucionar este tema.

| Código fuente

```
1  #include <iostream>
2  #include <cstring>
3  #include <cstdlib>
4  #include <string.h>
5
6  using namespace std;
7
8  const int MaxData=10;
9
10 struct Address{
11     char street[100];
12     char city[50];
13     char state[20];
14 };
15
16 struct persona{
17     char name[50];
18     struct Address Address_persona;
19     int codigo_postal;
20 }data[10];
21
22
23 int head=0;
24 int actual=0;
25 int ultimo=0;
26
27 void inserta(){
28     ultimo++;
29     fflush(stdin);
30     cout<<"Give the name: ";
```

```
31     cin.getline(data[ultimo].name,50);
32     cout<<"Give the street ";
33     cin.getline(data[ultimo].Address_persona.street,100);
34     cout<<"Give the city ";
35     cin.getline(data[ultimo].Address_persona.city,50);
36     cout<<"Estado: ";
37     cin.getline(data[ultimo].Address_persona.state,20);
38     cout<<"codigo postal: ";
39     scanf("%i",&data[ultimo].codigo_postal);
40 }
41
42 void siguiente(){
43     cout<<"\nName: "<<data[actual].name<<endl;
44     cout<<"street: "<<data[actual].Address_persona.street<<endl;
45     cout<<"City: "<<data[actual].Address_persona.city<<endl;
46     cout<<"State: "<<data[actual].Address_persona.state<<endl;
47     cout<<"pin: "<<data[actual].codigo_postal<<endl;
48     actual++;
49
50     if (actual >MaxData || actual>ultimo){
51         cout<<"ha llegado al ultimo nodo"<<endl;
52         system("pause");
53         actual--;
54     }
55 }
56
57 void imprimeLista(){
58     int i;
59     actual=head;
60
```

```
61     for(i=head; i<= ultimo; i ++)  
62         siguiente();  
63 }  
64 void buscar(int pos){  
65     cout<<"posicion: "<<pos<<endl;  
66     if (pos > ultimo || pos <head ){  
67         cout<<"posicion no encontrada"<<endl;  
68         return;  
69     }  
70     cout<<"\nName: "<<data[pos].name<<endl;  
71     cout<<"street: "<<data[pos].Address_persona.street<<endl;  
72     cout<<"City: "<<data[pos].Address_persona.city<<endl;  
73     cout<<"State: "<<data[pos].Address_persona.state<<endl;  
74     cout<<"pin: "<<data[pos].codigo_postal<<endl;  
75     pos++;  
76 }  
77  
78 void remover(){  
79     int indice;  
80     cout<<"Que nodo quieres remover?"<<endl; cin>>indice;  
81     indice--;  
82     if (indice > ultimo || indice <head ){  
83         cout<<"indice fuera de rango"<<endl;  
84         return;  
85     }  
86     int i;  
87     for(i=indice; i<=ultimo; i++){  
88         strcpy(data[i].name, data[i+1].name);  
89         strcpy(data[i].Address_persona.city, data[i+1].Address_persona.city);  
90         strcpy(data[i].Address_persona.street, data[i+1].Address_persona.street);
```

```
91     strcpy(data[i].Address_persona.state, data[i+1].Address_persona.state);
92     data[i].codigo_postal = data[i+1].codigo_postal;
93 }
94 if(indice == MaxData-1){
95     strcpy(data[indice].name,"");
96     strcpy(data[indice].Address_persona.city,"");
97     strcpy(data[indice].Address_persona.street,"");
98     strcpy(data[indice].Address_persona.state,"");
99     data[indice].codigo_postal=0;
100 }
101 ultimo--;
102 actual=head;
103 }
104
105 int main(){
106     int opcion, pos;
107     while(opcion!=6){
108         cout<<"Que opcion deseas, \n1-insertar \n2-imprime lista \n3-
109 siguiente nodo \n4-remove por nodo \n5-buscar por posicion \n6-salir
110 \nOpcion: "<<endl;
111         cin>>opcion;
112
113         switch(opcion){
114             case 1: system("cls");
115                     inserta();
116                     break;
117             case 2: system("cls");
118                     imprimeLista();
119                     break;
120             case 3: system("cls");
```

```
121             siguiente();
122             break;
123         case 4: system("cls");
124             remover();
125             system("pause");
126             break;
127         case 5: system("cls");
128             cout<<"Que posicion deseas consultar : "<<endl; cin>>pos;
129             buscar(pos);
130             break;
131         case 6: break;
132         default: printf("Esa opcion no existe...\n");
133     }
134 }
135
136 return 0;
137 }
```