

Actividad de Aprendizaje 02. La anidación estructural: Registros con Arreglos, Arreglos de Registros y Arreglos de Objetos

Christopher Ceballos Jiménez

219750442

Estructura de Datos I

```

2  #include <locale.h>
3  #include <string>
4
5  #include "date.h" DeleteP
6  #include "inventory.h"
7
8  using namespace std;
9
10 void RegistrationMenu(Inventory&);
11 void DeleteP(Inventory&);
12
13 int main()
14 {
15     setlocale(LC_ALL, "spanish");
16
17     char code[14];
18     Inventory InventoryProduct;
19
20     int op;
21
22     do
23     {
24         system("cls");
25         cout << "\n\n\t1) Registrar nuevo producto." << endl;
26         cout << "\t2) Remover de existencias." << endl;
27         cout << "\t3) Mostrar producto" << endl;
28         cout << "\t0) Salir" << endl;
29         cout << "Ingresa una opcion: "; cin >> op;
30
31         switch(op)
32         {
33             case 1:
34                 RegistrationMenu(InventoryProduct);
35                 cin.get();
36                 cout << "\nPresiona ENTER tecla para continuar";
37                 cin.get();
38                 break;
39             case 2:
40                 DeleteP(InventoryProduct);
41                 cin.get();
42                 cout << "\nPresiona ENTER tecla para continuar";
43                 cin.get();
44                 break;
45             case 3:
46                 cout << "\nIngresa un codigo 13(digitos): " << endl;
47                 cin >> code;
48                 InventoryProduct.showInformation(code);
49                 cin.get();
50                 cout << "\nPresiona ENTER tecla para continuar";
51                 cin.get();
52                 break;
53             case 0:
54                 cout << "Gracias por utilizar este programa !";
55                 break;
56             default:
57                 cout << "opcion NO valida !";58 }
59         }while(op!=0);
60
61
62     return 0;
63 }
64
65 void RegistrationMenu(Inventory &inventory)
66 {
67     char auxCode[14], auxName[30];
68     string aux;
69     float auxWeight, auxMostPrice, auxRetailPrice;
70     int auxStock;
71     Date auxDate;
72

```

```

73     cout << "\nIngresa un codigo 13(digitos): " << endl;
74     fflush(stdin);
75     cin.getline(auxCode,14,'\n');
76     if(inventory.validateCode(auxCode)==-1)
77     {
78         cout << "Ingresa el nombre del producto: " << endl;
79         cin.getline(auxName,29,'\n');
80
81         cout << "Ingresa el peso del producto: " << endl;
82         cin >> auxWeight;
83
84         cout << "Ingresa el precio de mayoreo: " << endl;
85         cin >> auxMostPrice;
86
87         cout << "Ingresa el precio al menudeo: " << endl;
88         cin >> auxRetailPrice;
89
90         cout << "Ingresa la existencia del producto: " << endl;
91         cin >> auxStock;
92
93         cout << "Ingresa el año de 'entrada': " << endl;
94         cin >> aux;
95         auxDate.setYear(aux);
96
97         cout << "Ingresa el mes de 'entrada': " << endl;
98         cin >> aux;
99         auxDate.setMonth(aux);
100
101         cout << "Ingresa el día de 'entrada': " << endl;
102         cin >> aux;
103         auxDate.setDay(aux);
104
105     inventory.registerNewProduct(auxCode,auxName,auxWeight,auxMostPrice,auxRetailPrice,auxStock,
auxDate);
106
107     }
108     else if (inventory.validateCode(auxCode)>=0)
109     {
110         int i = inventory.validateCode(auxCode);
111         cout << "¿Cuántos articulos nuevos hay? (Existencias) :" << endl;
112         cin >> auxStock;
113         inventory.addStock(auxStock,i);
114         cout << "\nLote de productos agregados correctamente! " << endl;
115     }
116     else
117         cout << "\nCodigo INVALIDO" << endl;
118 }
119
120
121 void DeleteP(Inventory &inventory)
122 {
123     char auxCode[14];
124     int amountRemove;
125     cout << "\nIngresa un codigo 13(digitos): " << endl;
126     cin >> auxCode;
127     cout << "¿Cuántos elementos deseas remover:" << endl;
128     cin >> amountRemove;
129     if(inventory.decreaseStock(amountRemove,inventory.validateCode(auxCode)))
130         cout << "\nProceso realizado CORRECTAMENTE ! , te quedan : " <<
inventory.getstock(inventory.validateCode(auxCode)) << " productos en el inventario" <<
endl;
131     else
132         cout << "No cuentas con esa cantidad de productos" << endl;
133 }
134

```

```

1  #ifndef DATE_H_INCLUDED
2  #define DATE_H_INCLUDED
3
4  #include <string>
5  #include <string.h>
6
7  class Date{
8  private:
9      std::string year;
10     std::string month;
11     std::string day;
12
13     bool validationYear(const std::string &);
14     bool validationMonth(const std::string &);
15     bool validationDay(const std::string &);
16
17 public:
18
19     Date(const std::string & = "0", const std::string & = "0", const std::string & =
"0");
20
21     void setYear(const std::string &);
22     void setMonth(const std::string &);
23     void setDay(const std::string &);
24
25     std::string getYear();
26     std::string getMonth();
27     std::string getDay();
28
29 };
30
31 #endif // DATE_H_INCLUDED

```

```

1  #include "date.h"
2
3  bool Date::validationYear(const std::string &y)
4  {
5      return (stoi(y)>=2010 && stoi(y)<=2021);
6  }
7
8  bool Date::validationMonth(const std::string &m)
9  {
10     return (stoi(m)>=1 && stoi(m)<=12);
11 }
12
13 bool Date::validationDay(const std::string &d)
14 {
15     return (stoi(d)>=1 && stoi(d)<=30);
16 }
17
18
19 Date::Date(const std::string &y , const std::string &m , const std::string &d)
20 {
21     year = validationYear(y) ? y : "2020";
22     month = validationMonth(m) ? m : "02";
23     day = validationDay(d) ? d : "07";
24 }
25
26 void Date::setYear(const std::string &y)
27 {
28     if (validationYear(y))
29         year = y;
30 }
31
32 void Date::setMonth(const std::string &m)
33 {

```

```

34     if (validationMonth(m))
35         month = m;
36
37 }
38
39 void Date::setDay(const std::string &d)
40 {
41     if(validationDay(d))
42         day = d;
43
44 }
45
46 std::string Date::getYear()
47 {
48     return year;
49 }
50
51 std::string Date::getMonth()
52 {
53     return month;
54 }
55
56 std::string Date::getDay()
57 {
58     return day;
59 }
60

```

```

1  #ifndef INVENTORY_H_INCLUDED
2  #define INVENTORY_H_INCLUDED
3
4  #include <iostream>
5
6  #include "date.h"
7
8  class Inventory{
9
10     private:
11
12         struct Product{
13             char code[14];
14             char name[30];
15             Date dateProduct;
16             float weight;
17             float mostrePrice;
18             float retailPrice;
19             int stock;
20         };
21
22         Product productos[500];
23
24         int registeredProducts=0;
25
26         int getLength(const char*);
27
28     public:
29
30         int validateCode(const char*);
31         void registerNewProduct(const char*, const char*,const float& , const
float&,const float& , const int&,Date&);

```

```

32
33     void addStock(const int&,const int&);
34     bool decreaseStock(const int& ,const int&);
35
36     int getstock(const int&);
37     void showInformation(const char*);
38 };
39
40 #endif // INVENTORY_H_INCLUDED

```

```

1  #include "inventory.h"
2
3  #include <iostream>
4
5  using namespace std;
6
7  int Inventory::getLength(const char *c)
8  {
9      int length=0;
10     for(int i(0) ; c[i]!='\0' ; i++)
11         length++;
12
13     return length;
14 }
15
16 int Inventory::validateCode(const char *c)
17 {
18     int i,j;
19
20     if(getLength(c)!=13)
21         return -2;
22
23     for(i=0; i<registeredProducts;i++)
24     {
25         for(j=0; j<13 ; j++)
26             if(c[j] != productos[i].code[j])
27                 break;
28
29         if(j==13)
30             return i;
31     }
32
33     return -1;
34 }
35
36 void Inventory::registerNewProduct(const char* c, const char* n,const float &w, const
float &mp,const float &rp, const int &s, Date&d)
37 {
38     for(int i=0; i<getLength(c);i++)
39         productos[registeredProducts].code[i] = c[i];
40
41     for(int i=0; i<getLength(n);i++)
42         productos[registeredProducts].name[i] = n[i];
43

```

```

44     productos[registeredProducts].weight = w;
45     productos[registeredProducts].mostrePrice = mp;
46     productos[registeredProducts].retailPrice = rp;
47     productos[registeredProducts].stock = s;
48     productos[registeredProducts].dateProduct.setYear(d.getYear());
49     productos[registeredProducts].dateProduct.setMonth(d.getMonth());
50     productos[registeredProducts].dateProduct.setDay(d.getDay()); 51
52     registeredProducts++;
53 }
54
55 void Inventory::addStock(const int&s ,const int &i)
56 {
57     productos[i].stock+=s;
58 }
59
60 bool Inventory::decreaseStock(const int&d ,const int &i)
61 {
62     if(productos[i].stock<d)
63         return false;
64
65     productos[i].stock-=d;
66     return true;
67 }
68
69
70 int Inventory::getstock(const int&i)
71 {
72     return productos[i].stock;
73 }
74
75 void Inventory::showInformation(const char*c)
76 {
77     int i=validateCode(c);
78
79     cout << "Nombre del producto: " << productos[i].name << endl;
80     cout << "Peso del producto: " << productos[i].weight << endl;
81     cout << "Precio de mayoreo: " << productos[i].mostrePrice << endl;
82     cout << "Precio al menudeo: " << productos[i].retailPrice << endl;
83     cout << "Existencia(s): " << productos[i].stock << endl;
84     cout << "Fecha de entrada: " << productos[i].dateProduct.getDay() << "/" <<
85     productos[i].dateProduct.getMonth() << "/" << productos[i].dateProduct.getYear();
86
87 }
88
89
90

```

```
"C:\Users\mbleo\OneDrive\Documentos\Estructura de datos 1\Tareas\Actividad 2\bin\Debug\Actividad 2.exe"

1) Registrar nuevo producto.
2) Remover de existencias.
3) Mostrar producto
0) Salir
Ingresa una opcion:
```

```
"C:\Users\mbleo\OneDrive\Documentos\Estructura de datos 1\Tareas\Actividad 2\bin\Debug\Actividad 2.exe"

1) Registrar nuevo producto.
2) Remover de existencias.
3) Mostrar producto
0) Salir
Ingresa una opcion: 1
Ingresa un codigo 13(digitos):
5555555555555
Ingresa el nombre del producto:
chocolate
Ingresa el peso del producto:
10
Ingresa el precio de mayoreo:
5
Ingresa el precio al menudeo:
10
Ingresa la existencia del producto:
20
Ingresa el año de 'entrada':
2020
Ingresa el mes de 'entrada':
02
Ingresa el día de 'entrada':
07
Presiona ENTER tecla para continuar
```



```
"C:\Users\mbleo\OneDrive\Documentos\Estructura de datos 1\Tareas\Actividad 2\bin\Debug\Actividad 2.exe"

1) Registrar nuevo producto.
2) Remover de existencias.
3) Mostrar producto
0) Salir
Ingresa una opcion: 3
Ingresa un codigo 13(digitos):
5555555555555
Nombre del producto: chocolate
Peso del producto: 10
Precio de mayoreo: 5
Precio al menudeo: 10
Existencia(s): 20
Fecha de entrada: 07/02/2020
Presiona ENTER tecla para continuar
```

```
"C:\Users\mbleo\OneDrive\Documentos\Estructura de datos 1\Tareas\Actividad 2\bin\Debug\Actividad 2.exe"

1) Registrar nuevo producto.
2) Remover de existencias.
3) Mostrar producto
0) Salir
Ingresa una opcion: 2
Ingresa un codigo 13(digitos):
5555555555555
¿Cuantos elementos deseas remover:
20

Proceso realizado CORRECTAMENTE ! , te quedan : 0 productos en el inventario
Presiona ENTER tecla para continuar
```