

```
1  /*
2  * Proyecto: STL-Map
3  * Archivo:  main.cpp
4  * Autor:    J. Miguel Guanira E.//miguel.guanira.
5  *
6  * Created on 19 de noviembre de 2024, 07:37 AM
7  */
8
9  #include <iostream>
10 #include <fstream>
11 #include <iomanip>
12 #include <map>
13 #include <iterator>
14 #include <string>
15 #include "Persona.h"
16 using namespace std;
17
18 int main(int argc, char** argv) {
19     // map<int,double>mapDlb;
20     // map<int,double>::iterator it;
21     // mapDlb[3]=45.87;
22     // mapDlb[8]=123.65;
23     // mapDlb[5]=63.75;
24     // mapDlb[17]=12.74;
25     // mapDlb[11]=756.54;
26     // mapDlb[8]=111.111;
27     //
28     //
29     // cout.precision(2);
30     // cout<<fixed;
31     //
32     // for(auto m: mapDlb)
33     //     cout<<setw(4)<<m.first<<setw(10)<<m.second<<endl;
34     // it = mapDlb.find(5);
35     // if(it!=mapDlb.end())
36     //     cout<<setw(4)<<it->first<<setw(10)<<it->second<<endl;
37     // else cout<<"No se encontro"<<endl;
38     // cout<<endl<<endl;
39     // mapDlb.erase(it);
40     // for(auto m: mapDlb)
41     //     cout<<setw(4)<<m.first<<setw(10)<<m.second<<endl;
42     //
43     // it = mapDlb.find(15);
44     // if(it!=mapDlb.end())
45     //     cout<<setw(4)<<it->first<<setw(10)<<it->second<<endl;
46     // else cout<<"No se encontro"<<endl;
47
48
49
50
51     // ifstream arch("Personal.csv",ios::in);
52     // if(not arch.is_open()){
53     //     cout<<"ERROR: No se pudo abrir el archivo "<<"Personal.csv"<<endl;
54     //     exit(1);
55     // }
56     // ofstream archRep("RepPersonal.txt",ios::out);
57     // if(not archRep.is_open()){
58     //     cout<<"ERROR: No se pudo abrir el archivo "<<"RepPersonal.txt"<<endl;
59     //     exit(1);
60     // }
61     //
62     // map<int,Persona> mapPer;
63     // class Persona per;
64     // map<int,Persona>::iterator itt;
65     //
66     // while(true) {
```

```

67 //      arch>>per;
68 //      if(arch.eof())break;
69 //      mapPer[per.GetDni()]=per;
70 //  }
71 //
72 //  for(auto m: mapPer)
73 //      archRep<<m.second;
74 //
75 //
76 //  int dni=514787;
77 //  itt = mapPer.find(dni);
78 //  archRep<<endl<<endl;
79 //
80 //  if(itt!=mapPer.end())
81 //      archRep<<itt->second;
82 //  else
83 //      archRep<<"No se encontro"<<endl;
84 //
85 //  dni=513387;
86 //  itt = mapPer.find(dni);
87 //
88 //  if(itt!=mapPer.end())
89 //      archRep<<itt->second;
90 //  else
91 //      archRep<<"No se encontro"<<endl;
92 //
93 //
94
95 map<string,Persona>mapPerStr;
96 ifstream arch("Personal.csv",ios::in);
97 if(not arch.is_open()){
98     cout<<"ERROR: No se pudo abrir el archivo "<<"Personal.csv"<<endl;
99     exit(1);
100 }
101 ofstream archRep("RepPersonalStr.txt",ios::out);
102 if(not archRep.is_open()){
103     cout<<"ERROR: No se pudo abrir el archivo "<<"RepPersonal.txt"<<endl;
104     exit(1);
105 }
106 class Persona per;
107 char nomb[60];
108 string nombre;
109 while(true){
110     arch>>per;
111     if(arch.eof())break;
112     per.GetNombre(nomb);
113     nombre = nomb;
114     mapPerStr[nombre]=per;
115 }
116 for(auto &m: mapPerStr){
117     archRep<<m.second;
118     m.second.SetDni(111);
119 }
120
121 for(auto m: mapPerStr){
122     archRep<<m.second;
123 }
124
125 for(auto &m: mapPerStr){
126     archRep<<m.second;
127     m.second.SetDni(111);
128 }
129
130
131 return 0;
132 }

```

```
133
134  /*
135   * Proyecto:  PlantillaDeClases
136   * Archivo:  Persona.h
137   * Autor:    J. Miguel GuaniraErazo (Juan Miguel)
138   *
139   * Creado el 30 de junio de 2020, 06:23 PM
140   */
141
142  #ifndef PERSONA_H
143  #define PERSONA_H
144  #include <fstream>
145  using namespace std;
146
147  class Persona {
148  private:
149      int dni;
150      char*nombre;
151      double sueldo;
152  public:
153      Persona();
154      Persona(const Persona& orig);
155      virtual ~Persona();
156      void SetSueldo(double sueldo);
157      double GetSueldo() const;
158      void SetNombre(const char* nombre);
159      void GetNombre(char* ) const;
160      void SetDni(int dni);
161      int GetDni() const;
162      void operator =(const class Persona&);
163      bool operator <(const class Persona&); //Obligatorio para ordenar
164  };
165  void operator >> (ifstream &, class Persona&);
166  ofstream& operator << (ofstream &,const class Persona&);
167  #endif /* PERSONA_H */
168
169  /*
170   * Proyecto:  PlantillaDeClases
171   * Archivo:  Persona.cpp
172   * Autor:    J. Miguel GuaniraErazo (Juan Miguel)
173   *
174   * Creado el 30 de junio de 2020, 06:23 PM
175   */
176  #include <iostream>
177  #include <fstream>
178  #include <iomanip>
179  using namespace std;
180  #include <cstring>
181  #include "Persona.h"
182
183  Persona::Persona() {
184      nombre = nullptr;
185  }
186
187  Persona::Persona(const Persona& orig) {
188      nombre = nullptr;
189      *this = orig;
190  }
191
192  Persona::~~Persona() {
193      if (nombre != nullptr) delete nombre;
194  }
195
196  void Persona::SetSueldo(double sueldo) {
197      this->sueldo = sueldo;
198  }
```

```
199
200     double Persona::GetSueldo() const {
201         return sueldo;
202     }
203
204     void Persona::SetNombre(const char* nomb) {
205         if (nombre != nullptr) delete nombre;
206         nombre = new char[strlen(nomb)+1];
207         strcpy(nombre,nomb);
208     }
209
210     void Persona::GetNombre(char*nomb) const {
211         if(nombre == nullptr) nomb[0]=0;
212         else strcpy(nomb,nombre);
213     }
214
215     void Persona::SetDni(int dni) {
216         this->dni = dni;
217     }
218
219     int Persona::GetDni() const {
220         return dni;
221     }
222
223     void Persona::operator =(const class Persona&per) {
224         char nomb[60];
225         dni = per.dni;
226         per.GetNombre(nomb);
227         SetNombre(nomb);
228         sueldo = per.sueldo;
229     }
230
231     bool Persona::operator<(const class Persona&orig) {
232         return strcmp(nombre,orig.nombre)>0;
233     }
234
235     void operator >> (ifstream &in, class Persona&per){
236         int dni;
237         char nomb[60];
238         double sueldo;
239         in>>dni;
240         in.get(); //coma
241         in.getline(nomb,60,',');
242         in>>sueldo;
243         per.SetDni(dni);
244         per.SetNombre(nomb);
245         per.SetSueldo(sueldo);
246     }
247
248     ofstream& operator << (ofstream &out,const class Persona&per){
249         out.precision(2);
250         out<<fixed;
251         char nomb[60];
252         per.GetNombre(nomb);
253         out<<left<<setw(10)<<per.GetDni()<<setw(40)<<nomb
254             <<right<<setw(10)<<per.GetSueldo()<<endl;
255         return out;
256     }
```