

```
1  /*
2   * Proyecto: STL-Vector
3   * Archivo:  main.cpp
4   * Autor:    J. Miguel Guanira E.
5   *
6   * Created on 12 de noviembre de 2024, 09:33 AM
7   */
8
9  #include <iostream>
10 #include <fstream>
11 #include <iomanip>
12 #include <vector>
13 #include <iterator>
14 #include <algorithm>
15 using namespace std;
16 #include "Persona.h"
17
18 int main(int argc, char** argv) {
19     /// vector <int> vectorInt;
20     /// cout<<"Numero de elementos: "<<vectorInt.size()<<endl;
21     /// cout<<"Capacidad          : "<<vectorInt.capacity()<<endl;
22     ///
23     /// vectorInt.push_back(56);
24     /// cout<<"Numero de elementos: "<<vectorInt.size()<<endl;
25     /// cout<<"Capacidad          : "<<vectorInt.capacity()<<endl;
26     ///
27     /// vectorInt.push_back(73);
28     /// cout<<"Numero de elementos: "<<vectorInt.size()<<endl;
29     /// cout<<"Capacidad          : "<<vectorInt.capacity()<<endl;
30     ///
31     /// vectorInt.push_back(109);
32     /// cout<<"Numero de elementos: "<<vectorInt.size()<<endl;
33     /// cout<<"Capacidad          : "<<vectorInt.capacity()<<endl;
34     ///
35     /// vectorInt.push_back(11);
36     /// vectorInt.push_back(45);
37     /// cout<<"Numero de elementos: "<<vectorInt.size()<<endl;
38     /// cout<<"Capacidad          : "<<vectorInt.capacity()<<endl;
39     ///
40     /// vectorInt.shrink_to_fit();
41     /// cout<<"Numero de elementos: "<<vectorInt.size()<<endl;
42     /// cout<<"Capacidad          : "<<vectorInt.capacity()<<endl;
43     /// for(int dato: vectorInt)
44     ///     cout<<setw(4)<<dato;
45     /// cout<<endl<<endl;
46     ///
47     /// for (int &dato:vectorInt)
48     ///     dato+=2;
49     ///
50     /// for(int dato: vectorInt)
51     ///     cout<<setw(4)<<dato;
52     /// cout<<endl<<endl;
53     ///
54     // vector <int> vect2 {34, 84 ,12, 55, 91, 101, 56, 3, 61, 74, 12};
55     /// cout<<"Numero de elementos: "<<vect2.size()<<endl;
56     /// cout<<"Capacidad          : "<<vect2.capacity()<<endl;
57     /// for(int dato: vect2)
58     ///     cout<<setw(4)<<dato;
59     /// cout<<endl<<endl;
60     /// for(int i=0; i<vect2.size(); i++)
61     ///     cout<<setw(4)<<vect2[i];
62     /// cout<<endl<<endl;
63     ///
64     /// vector <int> vect3 (10);
65     /// cout<<"Numero de elementos: "<<vect3.size()<<endl;
66     /// cout<<"Capacidad          : "<<vect3.capacity()<<endl;
```

```
67  /////    for(int dato: vect3)
68  /////        cout<<setw(4)<<dato;
69  /////    cout<<endl<<endl;
70  //
71  //    vector <int> vect4 (10,33);
72  /////    cout<<"Numero de elementos: "<<vect4.size()<<endl;
73  /////    cout<<"Capacidad          : "<<vect4.capacity()<<endl;
74  /////    for(int dato: vect4)
75  /////        cout<<setw(4)<<dato;
76  /////    cout<<endl<<endl;
77  /////
78  /////
79  /////    vect4.assign(5,66);
80  /////    for(int dato: vect4)
81  /////        cout<<setw(4)<<dato;
82  /////    cout<<endl<<endl;
83  /////    cout<<"Numero de elementos: "<<vect4.size()<<endl;
84  /////    cout<<"Capacidad          : "<<vect4.capacity()<<endl;
85  //
86  /////    vect4.assign({34, 84 ,12, 55, 91, 101, 56, 3, 61, 74, 12});
87  /////    for(int dato: vect4)
88  /////        cout<<setw(4)<<dato;
89  /////    cout<<endl<<endl;
90  /////    cout<<"Numero de elementos: "<<vect4.size()<<endl;
91  /////    cout<<"Capacidad          : "<<vect4.capacity()<<endl;
92  //
93  //    // vect4.assign(vect2);    //Esto no se puede hacer
94  /////    vect4 = vect2;
95  //    vect4 = {34, 84 ,12, 55, 91, 101, 56, 3, 61, 74, 12};
96  /////    for(int dato: vect4)
97  /////        cout<<setw(4)<<dato;
98  /////    cout<<endl<<endl;
99  /////    cout<<"Numero de elementos: "<<vect4.size()<<endl;
100  /////    cout<<"Capacidad          : "<<vect4.capacity()<<endl;
101  /////
102  /////    // elimina el último dato
103  /////    vect4.pop_back();
104  /////    for(int dato: vect4)
105  /////        cout<<setw(4)<<dato;
106  /////    cout<<endl<<endl;
107  /////    cout<<"Numero de elementos: "<<vect4.size()<<endl;
108  /////    cout<<"Capacidad          : "<<vect4.capacity()<<endl;
109  /////
110  /////    int dato1, dato2;
111  /////    dato1 = vect4.front();
112  /////    int &dato3 = vect4.front();
113  /////    dato2 = vect4.back();
114  /////    dato3 = 100;
115  /////    cout<<"Dato1: "<<dato1<<endl;
116  /////    cout<<"Dato2: "<<dato2<<endl;
117  /////    cout<<"Dato3: "<<dato3<<endl;
118  /////    for(int dato: vect4)
119  /////        cout<<setw(4)<<dato;
120  /////    cout<<endl<<endl;
121  /////    cout<<"Numero de elementos: "<<vect4.size()<<endl;
122  /////    cout<<"Capacidad          : "<<vect4.capacity()<<endl;
123  /////
124  /////    for(int dato: vect4)
125  /////        cout<<setw(4)<<dato;
126  /////    cout<<endl<<endl;
127  /////    cout<<"Numero de elementos: "<<vect4.size()<<endl;
128  /////    cout<<"Capacidad          : "<<vect4.capacity()<<endl;
129  //
130  //    //Iteradores
131  //
132  //    for(vector<int>::iterator it = vect4.begin(); it!=vect4.end(); it++ )
```

```

133 //      cout<<setw(4)<<*it;
134 //      cout<<endl<<endl;
135 //
136 //      vector<int>::iterator it2 = vect4.begin();
137 //      vector<int>::iterator it3 = vect4.begin();
138 //
139 //      it2+=3;
140 //      cout<<"IT2 = "<< *it2<<endl;
141 //      *it2 = 100;
142 //      for(int dato: vect4)
143 //          cout<<setw(4)<<dato;
144 //      cout<<endl<<endl;
145 //
146 //      vect4.clear();
147 //      cout<<"Numero de elementos: "<<vect4.size()<<endl;
148 //      cout<<"Capacidad          : "<<vect4.capacity()<<endl;
149 //
150 //      vect4.erase(it2);
151 //      it3 = vect4.begin() + 8;
152 //      vect4.erase(it2,it3);
153 //      for(int dato: vect4)
154 //          cout<<setw(4)<<dato;
155 //      cout<<endl<<endl;
156 //
157 //      cout<<"Numero de elementos: "<<vect4.size()<<endl;
158 //      cout<<"Capacidad          : "<<vect4.capacity()<<endl;
159 //
160 //
161 //      vect4.insert(it2,444);
162 //      vect4.insert(it2,3,444);
163 //      for(int dato: vect4)
164 //          cout<<setw(4)<<dato;
165 //      cout<<endl<<endl;
166 //
167 //      vect2.clear();
168 //      if(vect4.empty()) cout<<"V4 Vacio"<<endl;
169 //      else cout<<"V4 LLeno"<<endl;
170 //
171 //      if(vect2.empty()) cout<<"V2 Vacio"<<endl;
172 //      else cout<<"V2 LLeno"<<endl;
173 //
174 //      sort(vect4.begin(),vect4.end());
175 //      for(int dato: vect4)
176 //          cout<<setw(4)<<dato;
177 //      cout<<endl<<endl;
178 //
179 //      sort(it2,it3);
180 //      for(int dato: vect4)
181 //          cout<<setw(4)<<dato;
182 //      cout<<endl<<endl;
183 //
184
185 vector<class Persona> vectPer;
186 ifstream arch("Personal.csv",ios::in);
187 if(not arch.is_open()){
188     cout<<"ERROR: No se pudo abrir el archivo "<<"Personal.csv"<<endl;
189     exit(1);
190 }
191 ofstream archRep("RepPersonal.txt",ios::out);
192 if(not archRep.is_open()){
193     cout<<"ERROR: No se pudo abrir el archivo "<<"RepPersonal.txt"<<endl;
194     exit(1);
195 }
196 class Persona persona;
197
198 while(true){

```

```
199         arch>>persona;
200         if(arch.eof())break;
201         vectPer.push_back(persona);
202     }
203
204     sort(vectPer.begin(),vectPer.end());
205     for(Persona p: vectPer)
206         archRep<<p;
207
208     return 0;
209 }
210
211 /*
212  * Proyecto:  PlantillaDeClases
213  * Archivo:  Persona.h
214  * Autor:    J. Miguel GuaniraErazo (Juan Miguel)
215  *
216  * Creado el 30 de junio de 2020, 06:23 PM
217  */
218
219 #ifndef PERSONA_H
220 #define PERSONA_H
221 #include <iostream>
222 class Persona {
223 private:
224     int dni;
225     char*nombre;
226     double sueldo;
227 public:
228     Persona();
229     Persona(const Persona& orig);
230     virtual ~Persona();
231     void SetSueldo(double sueldo);
232     double GetSueldo() const;
233     void SetNombre(const char* nombre);
234     void GetNombre(char* ) const;
235     void SetDni(int dni);
236     int GetDni() const;
237     void operator =(const class Persona&);
238     bool operator <(const class Persona&); //OBLIGATORIO PARA ORDENAR
239 };
240 void operator >> (istream &, class Persona&);
241 ostream& operator << (ostream &,const class Persona&);
242 #endif /* PERSONA_H */
243
244
245
246 /*
247  * Proyecto:  PlantillaDeClases
248  * Archivo:  Persona.cpp
249  * Autor:    J. Miguel GuaniraErazo (Juan Miguel)
250  *
251  * Creado el 30 de junio de 2020, 06:23 PM
252  */
253 #include <iostream>
254 #include <iomanip>
255 using namespace std;
256 #include <cstring>
257 #include "Persona.h"
258
259 Persona::Persona() {
260     nombre = nullptr;
261 }
262
263 Persona::Persona(const Persona& orig) {
264     nombre = nullptr;
```

```
265     *this = orig;
266 }
267
268 Persona::~Persona() {
269     if (nombre != nullptr) delete nombre;
270 }
271
272 void Persona::SetSueldo(double sueldo) {
273     this->sueldo = sueldo;
274 }
275
276 double Persona::GetSueldo() const {
277     return sueldo;
278 }
279
280 void Persona::SetNombre(const char* nomb) {
281     if (nombre != nullptr) delete nombre;
282     nombre = new char[strlen(nomb)+1];
283     strcpy(nombre,nomb);
284 }
285
286 void Persona::GetNombre(char*nomb) const {
287     if(nombre == nullptr) nomb[0]=0;
288     else strcpy(nomb,nombre);
289 }
290
291 void Persona::SetDni(int dni) {
292     this->dni = dni;
293 }
294
295 int Persona::GetDni() const {
296     return dni;
297 }
298
299 void Persona::operator =(const class Persona&per) {
300     char nomb[60];
301     dni = per.dni;
302     per.GetNombre(nomb);
303     SetNombre(nomb);
304     sueldo = per.sueldo;
305 }
306
307 bool Persona::operator<(const class Persona&orig) {
308     return strcmp(nombre,orig.nombre)<0;
309 //     return strcmp(nombre,orig.nombre)>0;
310 //     return dni<orig.dni;
311 }
312
313 void operator >> (istream &in, class Persona&per){
314     int dni;
315     char nomb[60];
316     double sueldo;
317     in>>dni;
318     in.get(); //coma
319     in.getline(nomb,60,',');
320     in>>sueldo;
321     per.SetDni(dni);
322     per.SetNombre(nomb);
323     per.SetSueldo(sueldo);
324 }
325
326 ostream& operator << (ostream &out,const class Persona&per){
327     out.precision(2);
328     out<<fixed;
329     char nomb[60];
330     per.GetNombre(nomb);
```

```
331         out<<left<<setw(10)<<per.GetDni ()<<setw(40)<<nomb
332             <<right<<setw(10)<<per.GetSueldo ()<<endl;
333         return out;
334     }
```