

```
1  /*
2  * Proyecto: Preg02_Ex02_2022-1
3  * Archivo:  Pedido.h
4  * Autor:    J. Miguel Guanira E. //miguel.guanira.
5  *
6  * Created on 21 de noviembre de 2022, 07:56 AM
7  */
8
9
10 #ifndef PEDIDO_H
11 #define PEDIDO_H
12
13 class Pedido {
14 private:
15     int codigo;
16     int cantidad;
17     int dni;
18     int fecha;
19     double total;
20     int orden;
21 public:
22     void SetOrden(int orden);
23     int GetOrden() const;
24     void SetTotal(double total);
25     double GetTotal() const;
26     void SetFecha(int fecha);
27     int GetFecha() const;
28     void SetDni(int dni);
29     int GetDni() const;
30     void SetCantidad(int cantidad);
31     int GetCantidad() const;
32     void SetCodigo(int codigo);
33     int GetCodigo() const;
34     void imprime(ofstream &);
35     void dd_mm_aa(int &,int &, int &);
36 };
37
38 #endif /* PEDIDO_H */
39
40 /*
41 * Proyecto: Preg02_Ex02_2022-1
42 * Archivo:  Pedido.cpp
43 * Autor:    J. Miguel Guanira E. //miguel.guanira.
44 *
45 * Created on 21 de noviembre de 2022, 07:56 AM
46 */
47
48 #include <iostream>
49 #include <fstream>
50 #include <iomanip>
51 using namespace std;
52
53 #include "Pedido.h"
54
55 void Pedido::SetOrden(int orden) {
56     this->orden = orden;
57 }
58
59 int Pedido::GetOrden() const {
60     return orden;
61 }
62
63 void Pedido::SetTotal(double total) {
64     this->total = total;
```

```
65     }
66
67     double Pedido::GetTotal() const {
68         return total;
69     }
70
71     void Pedido::SetFecha(int fecha) {
72         this->fecha = fecha;
73     }
74
75     int Pedido::GetFecha() const {
76         return fecha;
77     }
78
79     void Pedido::SetDni(int dni) {
80         this->dni = dni;
81     }
82
83     int Pedido::GetDni() const {
84         return dni;
85     }
86
87     void Pedido::SetCantidad(int cantidad) {
88         this->cantidad = cantidad;
89     }
90
91     int Pedido::GetCantidad() const {
92         return cantidad;
93     }
94
95     void Pedido::SetCodigo(int codigo) {
96         this->codigo = codigo;
97     }
98
99     int Pedido::GetCodigo() const {
100         return codigo;
101     }
102
103     void Pedido::imprime(ofstream&arch) {
104         int dd,mm,aa;
105         dd_mm_aa(dd,mm,aa);
106         arch.precision(2);
107         arch<<fixed;
108         arch<<right<<setfill('0')<<setw(2)<<dd<<'/'<<setw(2)<<mm<<'/'
109             <<setw(4)<<aa<<setfill(' ');
110         arch<<setw(10)<<codigo<<setw(10)<<cantidad<<setw(10)<<total;
111         if(orden==0) arch<<setw(10)<<"No"<<endl;
112         else arch<<setw(10)<<"Si"<<endl;
113     }
114
115     void Pedido::dd_mm_aa(int &dd,int &mm, int &aa) {
116         int f = fecha;
117         aa = f/10000;
118         f %= 10000;
119         mm = f/100;
120         dd = f%100;
121     }
122
123     /*
124     * Proyecto: Preg02_Ex02_2022-1
125     * Archivo:  Producto.h
126     * Autor:    J. Miguel Guanira E. //miguel.guanira.
127     *
128     * Created on 21 de noviembre de 2022, 07:59 AM
```

```
129  */
130
131
132  #ifndef PRODUCTO_H
133  #define PRODUCTO_H
134
135  class Producto {
136  private:
137      int codprod;
138      char *nombre;
139      int stock;
140  public:
141      Producto();
142      Producto(const Producto& orig);
143      virtual ~Producto();
144      void SetStock(int stock);
145      int GetStock() const;
146      void SetNombre(const char* nombre);
147      void GetNombre(char*) const;
148      void SetCodprod(int codprod);
149      int GetCodprod() const;
150      void operator =(const Producto& orig);
151      virtual void leer(istream &);
152      virtual void imprime(ofstream &);
153      virtual int Prioridad()=0;
154  };
155
156  #endif /* PRODUCTO_H */
157
158  /*
159   * Proyecto: Preg02_Ex02_2022-1
160   * Archivo:  Producto.cpp
161   * Autor:    J. Miguel Guanira E. //miguel.guanira.
162   *
163   * Created on 21 de noviembre de 2022, 07:59 AM
164   */
165
166  #include <iostream>
167  #include <fstream>
168  #include <iomanip>
169  using namespace std;
170  #include <cstring>
171  #include "Producto.h"
172
173  Producto::Producto() {
174      nombre = nullptr;
175  }
176
177  Producto::Producto(const Producto& orig) {
178      nombre = nullptr;
179      *this = orig;
180  }
181
182  Producto::~~Producto() {
183      if(nombre!=nullptr) delete nombre;
184  }
185
186  void Producto::SetStock(int stock) {
187      this->stock = stock;
188  }
189
190  int Producto::GetStock() const {
191      return stock;
192  }
```

```

193
194 void Producto::SetNombre(const char* nomb) {
195     if(nombre != nullptr) delete nombre;
196     nombre = new char[strlen(nomb)+1];
197     strcpy(nombre,nomb);
198 }
199
200 void Producto::GetNombre(char*nomb) const {
201     if(nombre == nullptr) nomb[0] = 0;
202     else strcpy(nomb,nombre);
203 }
204
205 void Producto::SetCodprod(int codprod) {
206     this->codprod = codprod;
207 }
208
209 int Producto::GetCodprod() const {
210     return codprod;
211 }
212
213 void Producto::operator=(const Producto& orig) {
214     char cad[60];
215     codprod = orig.codprod;
216     orig.GetNombre(cad);
217     SetNombre(cad);
218     stock = orig.stock;
219 }
220
221 void Producto::leer(istream& arch) {
222     char cad[60],c;
223     arch>>codprod>>c;
224     arch.getline(cad,60,',');
225     SetNombre(cad);
226     arch>>stock;
227 }
228
229 void Producto::imprime(ofstream& arch) {
230     arch<<left<<setw(8)<<codprod<<setw(40)<<nombre<<right<<setw(10)<<stock;
231 }
232
233 /*
234  * Proyecto: Preg02_Ex02_2022-1
235  * Archivo: Categorical.h
236  * Autor: J. Miguel Guanira E. //miguel.guanira.
237  *
238  * Created on 21 de noviembre de 2022, 08:08 AM
239  */
240
241
242 #ifndef CATEGORIAL_H
243 #define CATEGORIAL_H
244 #include "Producto.h"
245 class Categorical : public Producto{
246 private:
247     int prioridad;
248     int minimo;
249 public:
250     void SetMinimo(int minimo);
251     int GetMinimo() const;
252     void SetPrioridad(int prioridad);
253     int GetPrioridad() const;
254     void leer(istream &);
255     void imprime(ofstream &);
256     int Prioridad();

```

```
257     };
258
259     #endif /* CATEGORIA1_H */
260
261     /*
262     * Proyecto: Preg02_Ex02_2022-1
263     * Archivo:  Categorical.cpp
264     * Autor:    J. Miguel Guanira E. //miguel.guanira.
265     *
266     * Created on 21 de noviembre de 2022, 08:08 AM
267     */
268
269     #include <iostream>
270     #include <fstream>
271     #include <iomanip>
272     using namespace std;
273
274     #include "Categorical.h"
275
276     void Categorical::SetMinimo(int minimo) {
277         this->minimo = minimo;
278     }
279
280     int Categorical::GetMinimo() const {
281         return minimo;
282     }
283
284     void Categorical::SetPrioridad(int prioridad) {
285         this->prioridad = prioridad;
286     }
287
288     int Categorical::GetPrioridad() const {
289         return prioridad;
290     }
291
292     void Categorical::leer(ifstream&arch) {
293         char c;
294         arch>>prioridad>>c>>minimo>>c;
295         Producto::leer(arch);
296     }
297
298     void Categorical::imprime(ofstream&arch) {
299         Producto::imprime(arch);
300         arch<<right<<setw(6)<<prioridad<<setw(10)<<minimo<<endl;
301     }
302
303     int Categorical::Prioridad() {
304         return prioridad;
305     }
306
307     /*
308     * Proyecto: Preg02_Ex02_2022-1
309     * Archivo:  Categoria2.h
310     * Autor:    J. Miguel Guanira E. //miguel.guanira.
311     *
312     * Created on 21 de noviembre de 2022, 08:11 AM
313     */
314
315
316     #ifndef CATEGORIA2_H
317     #define CATEGORIA2_H
318     #include "Producto.h"
319
320     class Categoria2 : public Producto{
```

```

321 private:
322     int prioridad;
323     double descuento;
324 public:
325     void SetDescuento(double descuento);
326     double GetDescuento() const;
327     void SetPrioridad(int prioridad);
328     int GetPrioridad() const;
329     void leer(istream &arch);
330     void imprime(ofstream &);
331     int Prioridad();
332 };
333
334 #endif /* CATEGORIA2_H */
335
336 /*
337  * Proyecto: Preg02_Ex02_2022-1
338  * Archivo: Categoria2.cpp
339  * Autor: J. Miguel Guanira E. //miguel.guanira.
340  *
341  * Created on 21 de noviembre de 2022, 08:11 AM
342  */
343
344 #include <iostream>
345 #include <fstream>
346 #include <iomanip>
347 using namespace std;
348
349 #include "Categoria2.h"
350
351 void Categoria2::SetDescuento(double descuento) {
352     this->descuento = descuento;
353 }
354
355 double Categoria2::GetDescuento() const {
356     return descuento;
357 }
358
359 void Categoria2::SetPrioridad(int prioridad) {
360     this->prioridad = prioridad;
361 }
362
363 int Categoria2::GetPrioridad() const {
364     return prioridad;
365 }
366
367 void Categoria2::leer(istream& arch) {
368     char c;
369     arch>>prioridad>>c>>descuento>>c;
370     Producto::leer(arch);
371 }
372
373 void Categoria2::imprime(ofstream&arch) {
374     arch.precision(2);
375     arch<<fixed;
376     Producto::imprime(arch);
377     arch<<right<<setw(6)<<prioridad<<setw(10)<<descuento<<endl;
378 }
379
380 int Categoria2::Prioridad() {
381     return prioridad;
382 }
383
384 /*

```

```
385  * Proyecto: Preg02_Ex02_2022-1
386  * Archivo:  Categoria3.h
387  * Autor:    J. Miguel Guanira E. //miguel.guanira.
388  *
389  * Created on 21 de noviembre de 2022, 08:14 AM
390  */
391
392
393  #ifndef CATEGORIA3_H
394  #define CATEGORIA3_H
395  #include "Producto.h"
396
397  class Categoria3 : public Producto{
398  private:
399      int prioridad;
400      double descuento;
401  public:
402      void SetDescuento(double descuento);
403      double GetDescuento() const;
404      void SetPrioridad(int prioridad);
405      int GetPrioridad() const;
406      void leer(istream &arch);
407      void imprime(ofstream &);
408      int Prioridad();
409  };
410
411  #endif /* CATEGORIA3_H */
412
413  /*
414   * Proyecto: Preg02_Ex02_2022-1
415   * Archivo:  Categoria3.cpp
416   * Autor:    J. Miguel Guanira E. //miguel.guanira.
417   *
418   * Created on 21 de noviembre de 2022, 08:14 AM
419   */
420
421  #include <iostream>
422  #include <fstream>
423  #include <iomanip>
424  using namespace std;
425
426  #include "Categoria3.h"
427
428  void Categoria3::SetDescuento(double descuento) {
429      this->descuento = descuento;
430  }
431
432  double Categoria3::GetDescuento() const {
433      return descuento;
434  }
435
436  void Categoria3::SetPrioridad(int prioridad) {
437      this->prioridad = prioridad;
438  }
439
440  int Categoria3::GetPrioridad() const {
441      return prioridad;
442  }
443
444  void Categoria3::leer(istream& arch) {
445      char c;
446      arch>>prioridad>>c>>descuento>>c;
447      Producto::leer(arch);
448  }
```

```
449
450 void Categoria3::imprime(ofstream&arch) {
451     arch.precision(2);
452     arch<<fixed;
453     Producto::imprime(arch);
454     arch<<right<<setw(6)<<prioridad<<setw(10)<<descuento<<endl;
455 }
456
457 int Categoria3::Prioridad() {
458     return prioridad;
459 }
460
461 /*
462  * Proyecto: Preg02_Ex02_2022-1
463  * Archivo: NProductos.h
464  * Autor: J. Miguel Guanira E. //miguel.guanira.
465  *
466  * Created on 21 de noviembre de 2022, 08:32 AM
467  */
468
469 #ifndef NPRODUCTOS_H
470 #define NPRODUCTOS_H
471 #include "Producto.h"
472 class NProductos {
473 private:
474     class Producto *prod;
475 public:
476     NProductos();
477     NProductos(const NProductos &orig);
478     void leeProducto(istream &);
479     void imprimeProducto(ofstream &);
480     void operator=(const NProductos &orig);
481     int GetCodigo();
482     int GetPrioridad();
483 };
484
485 #endif /* NPRODUCTOS_H */
486
487 /*
488  * Proyecto: Preg02_Ex02_2022-1
489  * Archivo: NProductos.cpp
490  * Autor: J. Miguel Guanira E. //miguel.guanira.
491  *
492  * Created on 21 de noviembre de 2022, 08:32 AM
493  */
494
495 #include <iostream>
496 #include <fstream>
497 #include <iomanip>
498 using namespace std;
499 #include "NProductos.h"
500 #include "Categorial.h"
501 #include "Categoria2.h"
502 #include "Categoria3.h"
503
504 NProductos::NProductos() {
505     prod = nullptr;
506 }
507
508 NProductos::NProductos(const NProductos &orig) {
509     *this = orig;
510 }
511
512
```



```

513 void NProductos::leeProducto(istream &arch) {
514     int cat;
515     arch>>cat;
516     if(arch.eof())return;
517     arch.get();
518     switch(cat){
519         case 1:
520             prod = new class Categoria1;
521             break;
522         case 2:
523             prod = new class Categoria2;
524             break;
525         case 3:
526             prod = new class Categoria3;
527             break;
528     }
529     prod->leer(arch); //Polimorfismo
530 }
531
532 void NProductos::imprimeProducto(ofstream&arch) {
533     prod->imprime(arch); //Polimorfismo
534 }
535
536 void NProductos::operator=(const NProductos& orig) {
537     prod = orig.prod;
538 }
539
540 int NProductos::GetCodigo() {
541     if(prod!=nullptr) return prod->GetCodprod();
542 }
543
544 int NProductos::GetPrioridad() {
545     if(prod!=nullptr) return prod->Prioridad();
546 }
547
548 /*
549  * Proyecto: Preg02_Ex02_2022-1
550  * Archivo:  Nodo.h
551  * Autor:    J. Miguel Guanira E. //miguel.guanira.
552  *
553  * Created on 21 de noviembre de 2022, 08:25 AM
554  */
555
556
557
558 #ifndef NODO_H
559 #define NODO_H
560 #include "Pedido.h"
561 #include "Lista.h"
562
563 class Nodo {
564 private:
565     class Pedido *ped;
566     class Nodo *sig;
567 public:
568     Nodo() {
569         sig = nullptr;
570     };
571     friend class Lista;
572 };
573
574 #endif /* NODO_H */
575
576 /*

```

```

577  * Proyecto: Preg02_Ex02_2022-1
578  * Archivo:  Lista.h
579  * Autor:    J. Miguel Guanira E. //miguel.guanira.
580  *
581  * Created on 21 de noviembre de 2022, 08:29 AM
582  */
583
584
585  #ifndef LISTA_H
586  #define LISTA_H
587  #include "Nodo.h"
588
589  class Lista {
590  private:
591      class Nodo *lini;
592      class Nodo *lfin;
593      class Pedido *leePedido(istream &);
594  public:
595      Lista();
596      virtual ~Lista();
597      void llenarLista(istream &);
598      void imprimeLista(ofstream &);
599      void obtenerCodigos(int *, int&);
600      void reordena(int *prior,int n);
601  };
602
603  #endif /* LISTA_H */
604
605  /*
606  * Proyecto: Preg02_Ex02_2022-1
607  * Archivo:  Lista.cpp
608  * Autor:    J. Miguel Guanira E. //miguel.guanira.
609  *
610  * Created on 21 de noviembre de 2022, 08:29 AM
611  */
612
613  #include <iostream>
614  #include <fstream>
615  #include <iomanip>
616  using namespace std;
617
618  #include "Lista.h"
619  #include "Categorial.h"
620
621  Lista::Lista() {
622      lini = nullptr;
623      lfin = nullptr;
624  }
625
626  Lista::~Lista() {
627      class Nodo *sale;
628      while(lini){
629          sale = lini;
630          lini = lini->sig;
631          delete sale;
632      }
633  }
634
635  void Lista::llenarLista(istream&arch) {
636      class Nodo*p, *ant, *nuevo;
637      class Pedido *ptProd;
638      while(true){
639          ptProd = leePedido(arch);
640          if (arch.eof())break;

```

```

641         nuevo= new class Nodo;
642         nuevo->ped = ptProd;
643         if (lini == nullptr){
644             lini = nuevo;
645             lfin = nuevo;
646         }
647         else {
648             p=lini;
649             ant=nullptr;
650             while(p) {
651                 if(p->ped->GetFecha()>ptProd->GetFecha())break;
652                 ant = p;
653                 p = p->sig;
654             }
655             nuevo->sig = p;
656             if(p==nullptr) lfin = nuevo;
657             if (ant != nullptr) ant->sig = nuevo;
658             else lini = nuevo;
659         }
660     }
661 }
662
663 Pedido* Lista::leePedido(istream&arch) {
664     int cod, cant,dni, fecha, dd,mm,aa;
665     double monto;
666     char c;
667     class Pedido *pt;
668
669     arch>>cod;
670     if(arch.eof())return nullptr;
671     arch>>c>>cant>>c>>monto>>c>>dni>>c>>dd>>c>>mm>>c>>aa;
672     pt = new class Pedido;
673     pt->SetCodigo(cod);
674     pt->SetCantidad(cant);
675     pt->SetTotal(monto);
676     pt->SetDni(dni);
677     fecha = 10000*aa+100*mm+dd;
678     pt->SetFecha(fecha);
679     return pt;
680 }
681
682 void Lista::imprimeLista(ofstream&arch) {
683     class Nodo*p=lini;
684     while(p) {
685         p->ped->imprime(arch);
686         p = p->sig;
687     }
688 }
689
690 void Lista::obtenerCodigos(int* codProd, int &n) {
691     class Nodo*p=lini;
692     n=0;
693     while(p) {
694         codProd[n] = p->ped->GetCodigo();
695         p = p->sig;
696         n++;
697     }
698 }
699
700 void Lista::reordena(int* prior, int n) {
701     class Nodo *p = lini->sig, *ant = lini;
702     int i=1;
703     while(p) {
704         if(prior[i]==1) {

```

```

705         p->ped->SetOrden(1);
706         ant->sig = p->sig;
707         p->sig = lini;
708         lini = p;
709         p=ant->sig;
710         if(p==nullptr) lfin = ant;
711     }
712     else {
713         p->ped->SetOrden(0);
714         ant = p;
715         p= p->sig;
716     }
717     i++;
718 }
719
720 }
721
722 /*
723  * Proyecto: Preg02_Ex02_2022-1
724  * Archivo: Programa.h
725  * Autor: J. Miguel Guanira E. //miguel.guanira.
726  *
727  * Created on 21 de noviembre de 2022, 08:35 AM
728  */
729
730
731 #ifndef PROGRAMA_H
732 #define PROGRAMA_H
733 #include "Lista.h"
734 #include <vector>
735 using namespace std;
736 #include "NProductos.h"
737 class Programa {
738 private:
739     class Lista lpedidos;
740     vector <class NProductos> vproductos;
741 public:
742     void carga();
743     void cargaProductos();
744     void cargaLista();
745     void muestra();
746     void ImprimeProductos();
747     void ImprimePedidos();
748     void actualiza();
749 };
750
751 #endif /* PROGRAMA_H */
752
753 /*
754  * Proyecto: Preg02_Ex02_2022-1
755  * Archivo: Programa.cpp
756  * Autor: J. Miguel Guanira E. //miguel.guanira.
757  *
758  * Created on 21 de noviembre de 2022, 08:35 AM
759  */
760
761 #include <iostream>
762 #include <iomanip>
763 #include <fstream>
764 using namespace std;
765
766 #include "Programa.h"
767 #include "NProductos.h"
768

```

```
769 void Programa::carga() {
770     cargaProductos();
771     cargaLista();
772 }
773
774 void Programa::cargaProductos() {
775     ifstream arch("Productos4.csv", ios::in);
776     if(not arch.is_open()){
777         cout<<"ERROR: No se pudo abrir el archivo Productos4.csv"<<endl;
778         exit(1);
779     }
780     int i=0;
781     class NProductos nprod;
782     while(true){
783         nprod.leeProducto(arch);
784         if(arch.eof())break;
785         vproductos.push_back(nprod);
786     }
787 }
788
789 void Programa::cargaLista() {
790     ifstream arch("Pedidos4.csv", ios::in);
791     if(not arch.is_open()){
792         cout<<"ERROR: No se pudo abrir el archivo Productos4.csv"<<endl;
793         exit(1);
794     }
795     lpedidos.llenarLista(arch);
796 }
797
798 void Programa::muestra() {
799     ImprimeProductos();
800     ImprimePedidos();
801 }
802
803 void Programa::ImprimeProductos() {
804     ofstream arch("ListaDeProductos.txt", ios::out);
805     if(not arch.is_open()){
806         cout<<"ERROR: No se pudo abrir el archivo ListaDeProductos.txt"<<endl;
807         exit(1);
808     }
809     class NProductos nprod;
810     for(int i=0; i<vproductos.size(); i++){
811         nprod = vproductos[i];
812         nprod.imprimeProducto(arch);
813     }
814 }
815
816 void Programa::ImprimePedidos() {
817     ofstream arch("ListaDePedidos.txt", ios::out);
818     if(not arch.is_open()){
819         cout<<"ERROR: No se pudo abrir el archivo ListaDePedidos.txt"<<endl;
820         exit(1);
821     }
822     lpedidos.imprimeLista(arch);
823 }
824
825 void Programa::actualiza() {
826     int codProd[200], prior[200], n;
827     class NProductos nprod;
828     lpedidos.obtenerCodigos(codProd,n);
829     for(int i=0; i<n; i++){
830         for(int k=0; k<vproductos.size();k++){
831             if(codProd[i] == vproductos[k].GetCodigo()){
832                 prior[i] = vproductos[k].GetPrioridad();
```

```
833         break;
834     }
835 }
836 }
837 lpedidos.reordena(prior, n);
838 }
839
840 /*
841  * Proyecto: Preg02_Ex02_2022-1
842  * Archivo:  main.cpp
843  * Autor:    J. Miguel Guanira E.//miguel.guanira.
844  *
845  * Created on 21 de noviembre de 2022, 07:53 AM
846  */
847
848 #include <iostream>
849 #include <iomanip>
850 using namespace std;
851 #include "Programa.h"
852 int main(int argc, char** argv) {
853     class Programa programa;
854
855     programa.carga();
856     programa.muestra();
857     programa.actualiza();
858     programa.muestra();
859
860     return 0;
861 }
862
```