```
1   /*
2    * Proyecto: SolucionLaboratorio09_2023_1
3    * Archivo:  main.cpp
4    * Autor:    J. Miguel Guanira E.
5    *
6    * Creado el 29 de mayo de 2024, 10:40 AM
7    */
8
9   #include <iostream>
10  #include <iomanip>
11  using namespace std;
12  #include "Tesoreria.h"
13  int main(int argc, char** argv) {
14      class Tesoreria caja;  // {}
15      caja.cargaEscalas("escalas.csv");
16      caja.cargaAlumnos("Alumnos.csv");
17      caja.actualizaBoleta();
18      caja.imprimeBoleta("reporteDePagos.txt");
19
20      return 0;
21  }
22
23  /*
24   * Proyecto: SolucionLaboratorio09_2023_1
25   * Archivo:  Tesoreria.h
26   * Autor:    J. Miguel Guanira E.
27   *
28   * Creado el 29 de mayo de 2024, 10:40 AM
29   */
30
31
32  #ifndef TESORERIA_H
33  #define TESORERIA_H
34  #include "Boleta.h"
35  #include "Escala.h"
36  class Tesoreria {
37  private:
38      class Boleta lboleta[100];
39      class Escala lescala[10];
40  public:
41      void cargaEscalas(const char *);
42      void cargaAlumnos(const char *);
43      void actualizaBoleta();
44      void imprimeBoleta(const char *);
45      void imprimeLinea(ofstream &, char, int);
46  };
47
48  #endif /* TESORERIA_H */
49
50  /*
51   * Proyecto: SolucionLaboratorio09_2023_1
52   * Archivo:  Tesoreria.h
53   * Autor:    J. Miguel Guanira E.
54   *
55   * Creado el 29 de mayo de 2024, 10:40 AM
56   */
57
58
59  #ifndef TESORERIA_H
60  #define TESORERIA_H
61  #include "Boleta.h"
62  #include "Escala.h"
63  class Tesoreria {
64  private:
65      class Boleta lboleta[100];
66      class Escala lescala[10];
```

```
67  public:
68      void cargaEscalas(const char *);
69      void cargaAlumnos(const char *);
70      void actualizaBoleta();
71      void imprimeBoleta(const char *);
72      void imprimeLinea(ofstream &, char, int);
73  };
74
75  #endif /* TESORERIA_H */
76
77  /*
78   * Proyecto: SolucionLaboratorio09_2023_1
79   * Archivo:  Tesoreria.cpp
80   * Autor:    J. Miguel Guanira E.
81   *
82   * Creado el 29 de mayo de 2024, 10:40 AM
83   */
84
85  #include <iostream>
86  #include <fstream>
87  #include <iomanip>
88  using namespace std;
89
90  #include "Tesoreria.h"
91
92  void Tesoreria::cargaEscalas(const char*nombArch) {
93      ifstream arch(nombArch, ios::in);
94      if(not arch.is_open()){
95          cout<<"No se pudo abrir el archivo "<<nombArch<<endl;
96          exit(1);
97      }
98      int cod;
99      double precio;
100
101     while(true){
102         arch>>cod;
103         if(arch.eof())break;
104         arch.get();
105         arch>>precio;
106         lescala[cod-1].SetCodigo(cod);
107         lescala[cod-1].SetPrecio(precio);
108     }
109 }
110
111 void Tesoreria::cargaAlumnos(const char*nombArch) {
112     ifstream arch(nombArch, ios::in);
113     if(not arch.is_open()){
114         cout<<"No se pudo abrir el archivo "<<nombArch<<endl;
115         exit(1);
116     }
117     char tipo;
118     int nD=0;
119     while(true){
120         arch>>tipo;
121         if(arch.eof())break;
122         arch.get();
123         lboleta[nD].asignaMemoria(tipo);
124         lboleta[nD].leeDatos(arch);
125         nD++;
126     }
127 }
128
129 void Tesoreria::actualizaBoleta() {
130     int escala;
131     double precioEscala;
132     for(int i=0; lboleta[i].hayDato(); i++){
```

```
133             escala = lboleta[i].GetEscala();
134             precioEscala = lescala[escala-1].GetPrecio();
135             lboleta[i].actualizaBoleta(precioEscala);
136         }
137     }
138
139     void Tesoreria::imprimeBoleta(const char*nombArch) {
140         ofstream arch(nombArch, ios::out);
141         if(not arch.is_open()){
142             cout<<"No se pudo abrir el archivo "<<nombArch<<endl;
143             exit(1);
144         }
145         arch<<left<<setw(10)<<"Codigo"<<setw(40)<<"Nombre"<<setw(8)<<"Escala"
146             <<setw(10)<<"Creditos"<<setw(12)<<"Licencia"<<setw(10)<<"Total"<<endl;
147         imprimeLinea(arch,'=',90);
148         for(int i=0; lboleta[i].hayDato(); i++){
149             lboleta[i].imprimeBoleta(arch);
150         }
151     }
152
153     void Tesoreria::imprimeLinea(ofstream&arch, char car, int nd) {
154         for(int i=0; i<nd; i++)
155             arch.put(car);
156         arch<<endl;
157     }
158
159     /*
160      * Proyecto: SolucionLaboratorio09_2023_1
161      * Archivo:  Boleta.h
162      * Autor:    J. Miguel Guanira E.
163      *
164      * Created on 19 de octubre de 2023, 07:19 PM
165      */
166
167
168     #ifndef BOLETA_H
169     #define BOLETA_H
170     #include <fstream>
171
172     #include "Alumno.h"
173
174     class Boleta {
175     private:
176         class Alumno *pboleta;
177     public:
178         Boleta();
179         virtual ~Boleta();
180         void asignaMemoria(char tipo);
181         void leeDatos(ifstream &arch) const;
182         bool hayDato();
183         int GetEscala();
184         void actualizaBoleta(double);
185         void imprimeBoleta(ofstream&);
186     };
187
188     #endif /* BOLETA_H */
189
190     /*
191      * Proyecto: SolucionLaboratorio09_2023_1
192      * Archivo:  Boleta.cpp
193      * Autor:    J. Miguel Guanira E.
194      *
195      * Created on 19 de octubre de 2023, 07:19 PM
196      */
197
198     #include <iostream>
```

```cpp
#include <iomanip>
using namespace std;
#include "Presencial.h"
#include "Semipresencial.h"
#include "Virtual.h"

#include "Boleta.h"

Boleta::Boleta() {
    pboleta = nullptr;
}

Boleta::~Boleta() {
    if(pboleta!=nullptr) delete pboleta;
}

void Boleta::asignaMemoria(char tipo) {
    switch(tipo){
        case 'P':
            pboleta = new class Presencial;
            break;
        case 'S':
            pboleta = new class Semipresencial;
            break;
        case 'V':
            pboleta = new class Virtual;
            break;
    }
}

void Boleta::leeDatos(ifstream& arch) const {
    pboleta->lee(arch);  // Aquí aplicamos Polimorfismo
}

bool Boleta::hayDato() {
    return pboleta != nullptr;
}

int Boleta::GetEscala() {
    return pboleta->GetEscala();
}

void Boleta::actualizaBoleta(double precioEscala) {
    pboleta->actualizaTotal(precioEscala);
}

void Boleta::imprimeBoleta(ofstream&arch) {
    pboleta->imprime(arch);
}


/*
 * Proyecto: SolucionLaboratorio09_2023_1
 * Archivo:  Alumno.h
 * Autor:    J. Miguel Guanira E.
 *
 * Creado el 29 de mayo de 2024, 10:40 AM
 */

#ifndef ALUMNO_H
#define ALUMNO_H

#include <fstream>

class Alumno {
private:
```

```
265         int codigo;
266         char *nombre;
267         int escala ;
268         double creditos;
269         double total;
270     public:
271         Alumno();
272         virtual ~Alumno();
273         void SetTotal(double total);
274         double GetTotal() const;
275         void SetEscala(int escala);
276         int GetEscala() const;
277         void SetNombre(const char* nombre);
278         void GetNombre(char*) const;
279         void SetCodigo(int codigo);
280         int GetCodigo() const;
281         void SetCreditos(double creditos);
282         double GetCreditos() const;
283         virtual void lee(ifstream &);
284         virtual void actualizaTotal(double);
285         virtual void imprime(ofstream&);
286     };
287
288     #endif /* ALUMNO_H */
289
290     /*
291      * Proyecto: SolucionLaboratorio09_2023_1
292      * Archivo:  Alumno.cpp
293      * Autor:    J. Miguel Guanira E.
294      *
295      * Creado el 29 de mayo de 2024, 10:40 AM
296      */
297
298     #include <iostream>
299     #include <fstream>
300     #include <iomanip>
301     using namespace std;
302     #include <cstring>
303     #include "Alumno.h"
304
305     Alumno::Alumno() {
306         nombre = nullptr;
307         total = 0;
308     }
309
310     Alumno::~Alumno() {
311         if(nombre!=nullptr) delete nombre;
312     }
313
314     void Alumno::SetTotal(double total) {
315         this->total = total;
316     }
317
318     double Alumno::GetTotal() const {
319         return total;
320     }
321
322     void Alumno::SetEscala(int escala) {
323         this->escala = escala;
324     }
325
326     int Alumno::GetEscala() const {
327         return escala;
328     }
329
330     void Alumno::GetNombre(char*cad) const {
```

```
331         if(nombre==nullptr) cad[0]=0;
332         else strcpy(cad,nombre);
333     }
334
335     void Alumno::SetNombre(const char* cad) {
336         if(nombre!=nullptr) delete nombre;
337         nombre = new char[strlen(cad)+1];
338         strcpy(nombre,cad);
339     }
340
341     void Alumno::SetCodigo(int codigo) {
342         this->codigo = codigo;
343     }
344
345     int Alumno::GetCodigo() const {
346         return codigo;
347     }
348
349
350     void Alumno::SetCreditos(double creditos) {
351         this->creditos = creditos;
352     }
353
354     double Alumno::GetCreditos() const {
355         return creditos;
356     }
357
358     void Alumno::lee(ifstream&arch) {
359         char nomb[60],c;
360         arch>>codigo;
361         arch.get();
362         arch.getline(nomb,60,',');
363         SetNombre(nomb);
364         arch>>escala>>c>>creditos;
365         arch.get();
366     }
367
368     void Alumno::actualizaTotal(double pago) {
369         total = pago;  // también   SetTotal(pago);
370     }
371
372     void Alumno::imprime(ofstream&arch) {
373         arch.precision(2);
374         arch<<fixed;
375         arch<<left<<setw(10)<<codigo<<setw(40)<<nombre<<right<<setw(3)<<escala
376             <<setw(12)<<creditos;
377
378     }
379
380     /*
381      * Proyecto: SolucionLaboratorio09_2023_1
382      * Archivo:  Presencial.h
383      * Autor:    J. Miguel Guanira E.
384      *
385      * Creado el 29 de mayo de 2024, 10:40 AM
386      */
387
388
389     #ifndef PRESENCIAL_H
390     #define PRESENCIAL_H
391     #include <fstream>
392     using namespace std;
393     #include "Alumno.h"
394
395     class Presencial : public Alumno{
396     private:
```

```
397         double recargo;
398         double total;
399     public:
400         Presencial();
401         void SetTotal(double total);
402         double GetTotal() const;
403         void SetRecargo(double recargo);
404         double GetRecargo() const;
405         void lee(ifstream &);
406         void actualizaTotal(double);
407         void imprime(ofstream&);
408     };
409
410     #endif /* PRESENCIAL_H */
411
412     /*
413      * Proyecto: SolucionLaboratorio09_2023_1
414      * Archivo:  Presencial.cpp
415      * Autor:    J. Miguel Guanira E.
416      *
417      * Creado el 29 de mayo de 2024, 10:40 AM
418      */
419
420     #include <iostream>
421     #include <iomanip>
422     using namespace std;
423
424     #include "Presencial.h"
425
426     Presencial::Presencial() {
427         total = 0.0;
428     }
429
430     void Presencial::SetTotal(double total) {
431         this->total = total;
432     }
433
434     double Presencial::GetTotal() const {
435         return total;
436     }
437
438     void Presencial::SetRecargo(double recargo) {
439         this->recargo = recargo;
440     }
441
442     double Presencial::GetRecargo() const {
443         return recargo;
444     }
445
446     void Presencial::lee(ifstream&arch) {
447         Alumno::lee(arch);
448         arch>>recargo;
449         arch.get(); // Saco el cambio de línea porque la línea empieza en
450                     // un caracter
451     }
452
453     void Presencial::actualizaTotal(double precioEscala) {
454         total = precioEscala*GetCreditos()*(1+GetRecargo()/100.0);
455         Alumno::SetTotal(total);
456     }
457
458     void Presencial::imprime(ofstream&arch) {
459         Alumno::imprime(arch);
460         arch<<setw(22)<<total<<endl;
461     }
462
```

```
463  /*
464   * Proyecto: SolucionLaboratorio09_2023_1
465   * Archivo:  Semipresencial.h
466   * Autor:    J. Miguel Guanira E.
467   *
468   * Creado el 29 de mayo de 2024, 10:40 AM
469   */
470
471
472  #ifndef SEMIPRESENCIAL_H
473  #define SEMIPRESENCIAL_H
474  #include <fstream>
475  using namespace std;
476  #include "Alumno.h"
477
478  class Semipresencial: public Alumno {
479  private:
480      double descuento;
481      double total;
482  public:
483      Semipresencial();
484      void SetTotal(double total);
485      double GetTotal() const;
486      void SetDescuento(double descuento);
487      double GetDescuento() const;
488      void lee(ifstream&);
489      void actualizaTotal(double);
490      void imprime(ofstream&);
491  };
492
493  #endif /* SEMIPRESENCIAL_H */
494
495  /*
496   * Proyecto: SolucionLaboratorio09_2023_1
497   * Archivo:  Semipresencial.cpp
498   * Autor:    J. Miguel Guanira E.
499   *
500   * Creado el 29 de mayo de 2024, 10:40 AM
501   */
502
503  #include <iostream>
504  #include <iomanip>
505  using namespace std;
506
507  #include "Semipresencial.h"
508
509  Semipresencial::Semipresencial() {
510      total = 0.0;
511  }
512
513  void Semipresencial::SetTotal(double total) {
514      this->total = total;
515  }
516
517  double Semipresencial::GetTotal() const {
518      return total;
519  }
520
521  void Semipresencial::SetDescuento(double descuento) {
522      this->descuento = descuento;
523  }
524
525  double Semipresencial::GetDescuento() const {
526      return descuento;
527  }
528
```

```
529  void Semipresencial::lee(ifstream&arch) {
530      Alumno::lee(arch);
531      arch>>descuento;
532      arch.get(); // Saco el cambio de línea porque la línea empieza en
533                  // un caracter
534  }
535
536  void Semipresencial::actualizaTotal(double precioEscala) {
537      total = precioEscala*GetCreditos()*(1-descuento/100.0);
538      Alumno::SetTotal(total);
539  }
540
541  void Semipresencial::imprime(ofstream&arch) {
542      Alumno::imprime(arch);
543      arch<<setw(22)<<total<<endl;
544  }
545
546  /*
547   * Proyecto: SolucionLaboratorio09_2023_1
548   * Archivo:  Virtual.h
549   * Autor:    J. Miguel Guanira E.
550   *
551   * Creado el 29 de mayo de 2024, 10:40 AM
552   */
553
554
555  #ifndef VIRTUAL_H
556  #define VIRTUAL_H
557  #include <fstream>
558  using namespace std;
559  #include "Alumno.h"
560
561  class Virtual: public Alumno {
562  private:
563      char *licencia;
564      double total;
565  public:
566      Virtual();
567      virtual ~Virtual();
568      void SetTotal(double total);
569      double GetTotal() const;
570      void SetLicencia(const char*);
571      void GetLicencia(char*) const;
572      void lee(ifstream&);
573      void actualizaTotal(double);
574      void imprime(ofstream&);
575  };
576
577  #endif /* VIRTUAL_H */
578
579  /*
580   * Proyecto: SolucionLaboratorio09_2023_1
581   * Archivo:  Virtual.cpp
582   * Autor:    J. Miguel Guanira E.
583   *
584   * Creado el 29 de mayo de 2024, 10:40 AM
585   */
586
587  #include <iostream>
588  #include <iomanip>
589  using namespace std;
590  #include <cstring>
591  #include "Virtual.h"
592
593  Virtual::Virtual() {
594      licencia = nullptr;
```

```
595         total = 0.0;
596     }
597
598     Virtual::~Virtual() {
599         if(licencia!=nullptr) delete licencia;
600     }
601
602     void Virtual::SetTotal(double total) {
603         this->total = total;
604     }
605
606     double Virtual::GetTotal() const {
607         return total;
608     }
609
610     void Virtual::SetLicencia(const char* cad) {
611         if(licencia!=nullptr) delete licencia;
612         licencia = new char[strlen(cad)+1];
613         strcpy(licencia,cad);
614     }
615
616     void Virtual::GetLicencia(char*cad) const {
617         if(licencia==nullptr) cad[0]=0;
618         else strcpy(cad,licencia);
619     }
620
621     void Virtual::lee(ifstream&arch) {
622         char lic[10];
623         Alumno::lee(arch);
624         arch>>lic;
625         arch.get();
626         SetLicencia(lic);
627     }
628
629     void Virtual::actualizaTotal(double precioEscala) {
630         total = precioEscala*GetCreditos() + 100.0;
631         Alumno::SetTotal(total);
632     }
633
634     void Virtual::imprime(ofstream&arch) {
635         Alumno::imprime(arch);
636         arch<<setw(12)<<licencia<<setw(10)<<total<<endl;
637     }
638
639     /*
640      * Proyecto: SolucionLaboratorio09_2023_1
641      * Archivo:  Escala.h
642      * Autor:    J. Miguel Guanira E.
643      *
644      * Creado el 29 de mayo de 2024, 10:40 AM
645      */
646
647     #ifndef ESCALA_H
648     #define ESCALA_H
649
650     class Escala {
651     private:
652         int codigo;
653         double precio;
654     public:
655         Escala();
656         void SetPrecio(double precio);
657         double GetPrecio() const;
658         void SetCodigo(int codigo);
659         int GetCodigo() const;
660     };
```

```
661
662 #endif /* ESCALA_H */
663
664 /*
665  * Proyecto: SolucionLaboratorio09_2023_1
666  * Archivo:  Escala.cpp
667  * Autor:    J. Miguel Guanira E.
668  *
669  * Creado el 29 de mayo de 2024, 10:40 AM
670  */
671
672 #include <iostream>
673 #include <iomanip>
674 using namespace std;
675
676 #include "Escala.h"
677
678 Escala::Escala() {
679     codigo = 0;
680 }
681
682 void Escala::SetPrecio(double precio) {
683     this->precio = precio;
684 }
685
686 double Escala::GetPrecio() const {
687     return precio;
688 }
689
690 void Escala::SetCodigo(int codigo) {
691     this->codigo = codigo;
692 }
693
694 int Escala::GetCodigo() const {
695     return codigo;
696 }
697
```