#include <fstream>

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

class Program

{

public:

string P\_Ad;

string version;

string P\_No;

string tarih;

void ProgramGir();

void ProgramYazdir();

};

void Program::ProgramGir()

{

cout.setf(ios::left);

cout<<setw(20)<<"Program Adi...:";

cin>>P\_Ad;

cout<<setw(20)<<"Program Versionu...:";

cin>>version;

cout<<setw(20)<<"Program No...:";

cin>>P\_No;

cout<<setw(20)<<"Program Tarih...:";

cin>>tarih;

}

void Program::ProgramYazdir()

{

cout.setf(ios::left);

cout<<setw(20)<<endl<<"Program Adi...:"<<P\_Ad<<endl;

cout<<setw(20)<<"Program Versionu...:"<<version<<endl;

cout<<setw(20)<<"Program No...:"<<P\_No<<endl;

cout<<setw(20)<<"Program Tarih...:"<<tarih<<endl;

}

class Musteri

{

public:

string M\_Adi;

string M\_SoyAdi;

string V\_tarih;

int guncelleme;

void MusteriGir();

void MusteriYaz();

};

void Musteri::MusteriGir()

{

cout.setf(ios::left);

cout<<setw(20)<<"Musteri Adi...:";

cin>>M\_Adi;

cout<<setw(20)<<"Musteri Soyadi...:";

cin>>M\_SoyAdi;

cout<<setw(20)<<"Verilis Tarihi...:";

cin>>V\_tarih;

cout<<setw(20)<<"Guncelleme Sayisi:";

cin>>guncelleme;

}

void Musteri::MusteriYaz()

{

cout.setf(ios::left);

cout<<setw(20)<<endl<<"Musteri Adi...:"<<M\_Adi<<endl;

cout<<setw(20)<<"Musteri Soyadi...:"<<M\_SoyAdi<<endl;

cout<<setw(20)<<"Verilis Tarihi...:"<<V\_tarih<<endl;

cout<<setw(20)<<"Program güncellemesi...:"<<guncelleme<<endl;

}

class Kayitlar

{

public:

Program Programlar[50];

Musteri Musteriler[50];

Program Temp\_Program;

Program Temp\_Program2;

Program Temp\_Program3;

Musteri Temp\_Musteri;

string nu;

string nu2;

int aa;

int ProgSayisi;

int MustSayisi;

string temp\_no; // geçici değişkeni tutmak için, silmede 2 kere yazmaması için

fstream dosya1;

fstream dosya2;

fstream temp\_dosya;

Kayitlar() // constractor kurulumu

{

ProgSayisi=0;

MustSayisi=0;

aa=0;

/\*dosya1.open("program kayitlari.txt",ios::out|ios::in|ios::app);

dosya1.setf(ios::left);

dosya2.open("musteri kayitlari.txt",ios::out|ios::in|ios::app);

dosya2.setf(ios::left);\*/

}

void ProgramEkle()

{

dosya1.open("program kayitlari.txt",ios::out|ios::in|ios::app);

dosya1.setf(ios::left);

Programlar[ProgSayisi].ProgramGir();

Program P = Programlar[ProgSayisi]; // kolaylık için P'yi Programlar[ProgSayisi]'na eşitledim

dosya1

<<setw(20)<<P.P\_Ad

<<setw(20)<<P.version

<<setw(20)<<P.P\_No

<<setw(20)<<P.tarih

<<endl;

dosya1.close();

}

bool ProgramSec(string Numara)

{

dosya1.open("program kayitlari.txt",ios::out|ios::in|ios::app);

dosya1.setf(ios::left);

while(!dosya1.eof())

{

dosya1>>Temp\_Program.P\_Ad;

dosya1>>Temp\_Program.version;

dosya1>>Temp\_Program.P\_No;

dosya1>>Temp\_Program.tarih;

if(Temp\_Program.P\_No==Numara)

{

Temp\_Program.ProgramYazdir();

return false;

}

if(Temp\_Program.P\_No!=Numara && dosya1.eof())

{

cout<<"Program bulunamadi"<<endl;

return true;

}

}

}

void MusteriEkle()

{

cout<<"Satilan Program Numarasini Giriniz:";

dosya1.close();

cin>>nu;

bool bProgram= ProgramSec(nu);

if(bProgram)

{

dosya1.close();

return MusteriEkle();

}

dosya1.close();

Musteriler[MustSayisi].MusteriGir();

Musteri M = Musteriler[MustSayisi]; // kolaylık için M'yi Müşterilere eşitledim

dosya2.open("musteri kayitlari.txt",ios::out|ios::in|ios::app);

dosya2.setf(ios::left);

dosya2

<<setw(20)<<M.M\_Adi

<<setw(20)<<M.M\_SoyAdi

<<setw(20)<<M.guncelleme

<<setw(20)<<M.V\_tarih

<<setw(20)<<Temp\_Program.P\_Ad

<<setw(20)<<Temp\_Program.version

<<setw(20)<<Temp\_Program.P\_No

<<setw(20)<<Temp\_Program.tarih

<<endl;

dosya2.close();

}

void ProgramGoster()

{

temp\_no="0";

dosya1.open("program kayitlari.txt",ios::out|ios::in);

while(!dosya1.eof())

{

dosya1>>Temp\_Program3.P\_Ad;

dosya1>>Temp\_Program3.version;

dosya1>>Temp\_Program3.P\_No;

dosya1>>Temp\_Program3.tarih;

if(temp\_no!=Temp\_Program3.P\_No)

{

Temp\_Program3.ProgramYazdir();

}

temp\_no=Temp\_Program3.P\_No;

}

dosya1.close();

}

void MusteriGoster()

{

dosya2.open("musteri kayitlari.txt",ios::out|ios::in);

temp\_no="0";

while(!dosya2.eof())

{

dosya2>>Temp\_Musteri.M\_Adi;

dosya2>>Temp\_Musteri.M\_SoyAdi;

dosya2>>Temp\_Musteri.guncelleme;

dosya2>>Temp\_Musteri.V\_tarih;

dosya2>>Temp\_Program3.P\_Ad;

dosya2>>Temp\_Program3.version;

dosya2>>Temp\_Program3.P\_No;

dosya2>>Temp\_Program3.tarih;

if(temp\_no!=Temp\_Program3.P\_No)

{

Temp\_Musteri.MusteriYaz();

Temp\_Program3.ProgramYazdir();

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

}

temp\_no=Temp\_Program3.P\_No;

}

dosya2.close();

}

void ProgramSil()

{

dosya1.open("program kayitlari.txt",ios::out|ios::in|ios::app);

temp\_dosya.open("gecici dosya.txt",ios::out|ios::in|ios::app);

temp\_dosya.setf(ios::left);

temp\_no="0";

cout<<"Silinecek programin numarasini giriniz...:";

cin>>nu2;

while(!dosya1.eof())

{

dosya1>>Temp\_Program2.P\_Ad;

dosya1>>Temp\_Program2.version;

dosya1>>Temp\_Program2.P\_No;

dosya1>>Temp\_Program2.tarih;

if(Temp\_Program2.P\_No==nu2)

{

Temp\_Program2.ProgramYazdir();

aa=1;

}

if(Temp\_Program2.P\_No!=nu2 && temp\_no!=Temp\_Program2.P\_No)

{

temp\_dosya<<setw(20)<<Temp\_Program2.P\_Ad

<<setw(20)<<Temp\_Program2.version

<<setw(20)<<Temp\_Program2.P\_No

<<Temp\_Program2.tarih<<endl;

}

if(aa==0 && dosya1.eof())

{

cout<<"Program bulunamadi"<<endl;

aa=0;

}

temp\_no=Temp\_Program2.P\_No;

}

dosya1.close();

temp\_dosya.close();

remove("program kayitlari.txt");

rename("gecici dosya.txt","program kayitlari.txt");

// emin misiniz ekle! (e/h) !

}

void MusteriSil()

{

dosya2.open("musteri kayitlari.txt",ios::out|ios::in|ios::app);

temp\_dosya.open("gecici dosya.txt",ios::out|ios::in|ios::app);

temp\_dosya.setf(ios::left);

temp\_no="0";

cout<<"Silinecek programin numarasini giriniz...:";

cin>>nu2;

while(!dosya2.eof())

{

dosya2>>Temp\_Musteri.M\_Adi;

dosya2>>Temp\_Musteri.M\_SoyAdi;

dosya2>>Temp\_Musteri.guncelleme;

dosya2>>Temp\_Musteri.V\_tarih;

dosya2>>Temp\_Program2.P\_Ad;

dosya2>>Temp\_Program2.version;

dosya2>>Temp\_Program2.P\_No;

dosya2>>Temp\_Program2.tarih;

if(Temp\_Program2.P\_No==nu2)

{

Temp\_Musteri.MusteriYaz();

Temp\_Program2.ProgramYazdir();

aa=1;

}

if(Temp\_Program2.P\_No!=nu2 && temp\_no!=Temp\_Program2.P\_No)

{

temp\_dosya<<setw(20)<<Temp\_Musteri.M\_Adi

<<setw(20)<<Temp\_Musteri.M\_SoyAdi

<<setw(20)<<Temp\_Musteri.guncelleme

<<setw(20)<<Temp\_Musteri.V\_tarih

<<setw(20)<<Temp\_Program2.P\_Ad

<<setw(20)<<Temp\_Program2.version

<<setw(20)<<Temp\_Program2.P\_No

<<Temp\_Program2.tarih<<endl;

}

if(aa==0 && dosya2.eof())

{

cout<<"Musteri bulunamadi"<<endl;

aa=0;

}

temp\_no=Temp\_Program2.P\_No;

}

dosya2.close();

temp\_dosya.close();

remove("musteri kayitlari.txt");

rename("gecici dosya.txt","musteri kayitlari.txt");

}

};

int main()

{

int islem=0;

Kayitlar Kayit1;

cout<<"...::: Dosyalama islemleri :::..."<<endl;

while(1)

{

cout<<endl<<endl;

cout<<endl<<"1.Program Kaydet"<<endl<<"2.Musteri Kaydet"<<endl<<"3.Program Sil"<<endl<<"4.Musteri Sil"<<endl<<"5.Programlari Listele"

<<endl<<"6.Musteri Goster"<<endl<<"7.Cikis";

cout<<endl<<"Yapmak istediginiz islemin numarasini giriniz...";

etiket1:

cin>> islem;

switch(islem)

{

case 1: Kayit1.ProgramEkle(); break;

case 2: Kayit1.MusteriEkle(); break;

case 3: Kayit1.ProgramSil(); break;

case 4: Kayit1.MusteriSil(); break;

case 5: Kayit1.ProgramGoster(); break;

case 6: Kayit1.MusteriGoster(); break;

case 7: goto etiket2; break;

default:

{

cout<<"Yanlis Numara girdiniz...!! Tekrar giriniz";

goto etiket1;

} break;

}

}

etiket2:

cout<<"Program kapaniyor...";

}