**Algoritma ve Programlama II**

Proje 2  
Havayolu Bilet Satış Sistemi

05100000003 - Abdulbaki Furkan TANRIVERDİ

05100000007 – Burak YABAR

05110000064 – Onur Cem ŞENEL

**Teslim Tarihi:** 21.03.2012

ANALİZ

Bizden nesneye yönelik programlama yöntemi ile, bir havayolu şirketinin günlük uçuşlarının takip edilmesini sağlayacak bir uygulama geliştirilmesi istenmektedir. Bu amaçla uçuşlara ilişkin aşağıdaki bilgilere, hem uçuş numaraları hem kalkış yerleri aracılığıyla hızlı erişilebilmesi istenmektedir:

Uçuşnumarası :Tamsayı

Kalkış yeri : 20 karakter uzunluğunda string (sadece latin harfleri)

Varış yeri : 20 karakteruzunluğunda string (sadece latin harfleri)

Kalkış zamanı : 4 karakter uzunluğunda string

Normal koltuk sayısı : Tamsayı

Lüks koltuk sayısı : Tamsayı

Uçuş bilgilerine uçuş numarası aracılığıyla erişmek için, uçuş numarasına göre küçükten büyüğe doğru sıralı olan bir tek-bağlı liste kullanacağız.Uçuş bilgilerine kalkış yeri aracılığıyla erişmek için ise kalkış yerine göre (kalkış yeri aynı ise kalkış zamanına göre) sıralı olan bir çift-bağlı liste kullanacağız.

Programdaki en kritik nokta, karakter dizileri kullanmak yerine, bizden dokümanda da istenildği gibi **string** sıınıfı oluşturmak ve buradan nesneler türetmek. Ayrıca C++ ‘ ın önemli konularından olan **overloading** te string sınıfından oluşturulan nesnelerin karşılaştırılması için kullanılacak. Tabiki overload fonksiyonlarının içinde bize **<cstring>** sınıfının hazır fonksiyonlarının kullanım hakkı verilmiş.

İstenilenleri kullanıcıya bir menü aracılığıyla sunacağız.

TASARIM

Programımızda birden fazla kavram olduğu için, bu kavramları ifade etmek için herbiri için ayrı **class** yaptık. Bu classların veri elemanları **private** , çağrılan metotları ise **public** olarak tanımladık.

Programda Kullanılan Sınıflar ve Metotlar

**(1) class Kalkis\_Yerine\_Gore\_Ucus\_Listesi { friend** ostream &operator<<(ostream &,Kalkis\_Yerine\_Gore\_Ucus\_Listesi &); **private:**  Ucus listeBasi; **public:**  Kalkis\_Yerine\_Gore\_Ucus\_Listesi();void ekle(Ucus \*); Ucus \*ara(int);void ara(String,int); void ara(String,String); void cikar(Ucus \*); **};**

Bizden uçuşların kalkış yerine göre(kalkış yeri aynı olan uçuşların ise kalkış zamanına göre) çift bağlı liste ile bellekte dinamik olarak tutulması istenmektedir. Bu yüzden bu liste için bir sınıf oluşturduk. Bizim sınıfımıza ait olamayan **ostream** sınıfından **<<**operatörünü overload yapmak için bu sınıfı **friend** olarak ayarlayıp sınıfımızda kullandık. **Private** olarak tanımladığımız eleman **UcuslisteBasi** yani uçuş düğümlerinin ilk elemanıdır.**Public** olarak ise, yapıcı metot olan **Kalkis\_Yerine\_Gore\_Ucus\_Listesi()**, uçuş eklemek için **void ekle(Ucus \*),** uçuş olup olmadığının kontrolünü uçuş numarasına göre yapan metot için **Ucus \*ara(int),** sadece kalkış yerine ya da sadece varış yerine göre arama yapan metot için **void ara(String,int),** hem kalkış yerine hem de varış yerine göre arama yapan metot için overload ettiğimiz **void ara(String,String)** ve düğümden uçuş çıkarmak için **void cikar(Ucus \*)** metotlarını tanımladık.

**(2) class String { friend** ostream &operator<<(ostream &,const String &); **friend** istream &operator>>(istream &,String &); **private:**  int boyut; char \*ptr; **public:**  String(int=21); bool operator==(const String &); bool operator>(const String &); bool operator>=(const String &); **};**

Bizden kalkış ve varış yerleri için karakter dizisi kullanmak yerine **string** sınıfı oluşturmamız bekleniyor. Ayrıca çıktı için **<<**, girdi için **>>**, eşitlik kontrolü için **==**, alfabetik sıraya göre büyüklük-küçüklük için **>**, ayrıca eşitlik için **>=** operatörlerini overload ettik. Yine sınıfımıza ait olmayan **ostream** ve **istream** sınıflarının metotlarını kullanmak için onları **friend** olarak tanımladık. Girilen karakter dizisinin eleman sayısı için **boyut** ve karakter dizisinin adresini tutan **\*ptr** veri elemanlarını **private** olarak tanımladık. **Public** olarak, yapıcı metot olan ve karakter dizisine maksimum boyut olarak nesne oluşturulduğu anda 21 elemanlık alan tanıyan **String(int=21),**dizilerin eşitliğinin karşılaştırılması için**bool operator==(const String &),** büyüklük ve eşitlik karşılaştırmaları için**bool operator>(const String &)** ve **bool operator>=(const String &)**metotlarını kullandık.

**(3) class Ucus { friend** ostream &operator<<(ostream &,Ucus &); **private:** const int ucusNo; const String kalkisYeri; const String varisYeri; const String zaman; const int normalKoltukSayisi; int luksKoltukSayisi; Ucus \*onceki; Ucus \*sonraki; Ucus \*cSonraki; **public:** Ucus(const int,const String,const String,const String,const int,int,Ucus \* = NULL,Ucus \* = NULL,Ucus \* = NULL); int ucusNoOku(); String kalkisYeriOku(); String varisYeriOku(); String zamanOku(); int normalKoltukOku(); Int luksKoltukOku(); void luksKoltukAyarla(int); Ucus \*oncekiOku(); void oncekiAyarla(Ucus \*); Ucus \*sonrakiOku(); void sonrakiAyarla(Ucus \*); Ucus \*cSonrakiOku(); void cSonrakiAyarla(Ucus \*); **};**

Programımızda uçuşların bilgilerinin tutulabilmesi, değişiklik yapılabilmesi ya da ekrana yazdırılabilmesi için bir sınıf oluşturmamız beklenmektedir. Bizde bunun için **class Ucus** isimli birsınıf tanımladık. Bizim sınıfımıza ait olmayan **ostream** sınıfından **<<** operatörünü overload yapmak için bu sınıfı **friend** olarak çağırıp sınıfımızda kullandık.**Private** veri elemanlarında ise **const** olarak ayarladığımız (değeri değişmeyecek elemanlar için) **ucusNo,** kalkış yerini tutmak için **kalkisYeri,** varış yerini tutmak için **varisYeri,** zamanını tutmak için **zaman** ve normal koltuk sayısını tutmak için **normalKoltukSayisi** değişkenleri ile değeri değişebilen bir very elemanı olan lüks koltuk sayısını tutmak için **luksKoltukSayisi** değişkenlerini kullanldık. Programımızda tek bağlı ve çift bağlı listelerde kullanılacağı için bu düğümlerin adreslerini tutmaları için **\*onceki**, **\*sonraki(**çift bağlı liste için**)**  ve **\*cSonraki(**tek bağlı liste**)** işaretçilerini tanımladık. **Public** eleman olarak ise yapıcı metodumuz olarak **Ucus(const int,const String,const String,const String,const int,int,Ucus \* = NULL,Ucus \* = NULL,Ucus \* = NULL)** metodu, nesnenin uçuş numarasını geri döndüren **ucusNoOku()**, kalkış yerini döndüren **kalkisYeriOku()**, varış yerini döndüren **varisYeriOku()**, zamanını döndüren **zamanOku()**, normal koltuk sayısını döndüren **normalKoltukOku()**, lüks koltuk sayısını döndüren **luksKoltukOku()** metodları ile lüks koltuk sayısı değiştirilmek istendiğinde girilen değeri yeni lüks koltuk sayısı olarak ayarlayan **luksKoltukAyarla(int)** metodudur. Ayrıca tek bağlı ve ya çift bağlı listelerde bağ sahalarının okunması için **\*oncekiOku(), \*sonrakiOku()\*cSonrakiOku()** metodları ile bağ sahalarının değiştirilmesi için **oncekiAyarla(Ucus \*), sonrakiAyarla(Ucus \*), cSonrakiAyarla(Ucus \*)** metodlarını da **public** eleman olarak tanımladık.

**(4)** **class Ucus\_Numarasina\_Gore\_Ucus\_Listesi {** **friend** ostream &operator<<(ostream &,const Ucus\_Numarasina\_Gore\_Ucus\_Listesi &); **private:** Ucus \*ilk; **public:** Ucus\_Numarasina\_Gore\_Ucus\_Listesi(); void ekle(Ucus \*); Ucus \*ara(int); void cikar(Ucus \*); void luksListele(); **};**

Bizden uçuşların uçus numarasına göre küçükten büyüğe doğru sıralı olan bir tek bağlı liste ile bellekte tutulması istenmektedir. Bu yüzden bu liste için bir ayrı bir sınıf oluşturduk. Bizim sınıfımıza ait olamayan **ostream** sınıfından **<<**operatörünü overload yapmak için bu sınıfı **friend** olarak tanımlayıp sınıfımızda kullandık. **Private** olarak tanımladığımız eleman \***ilk** uçuş sınıfından bir işaretçidir yani uçuş düğümlerinin ilk elemanının adresini tutmaktadır. **Public** olarak ise, yapıcı metot olan **Ucus\_Numarasina\_Gore\_Ucus\_Listesi()**, uçuş eklemek için **void ekle(Ucus\*)**,uçuş olup olmadığının kontrolünü uçuş numarasına göre yapmak için **Ucus \*ara(int),** düğümden uçuş çıkarmak için **void cikar(Ucus \*)** ve lüks koltuklara sahip olan uçuşları listelemek için **void luksListele()** metotlarını tanımladık.

**PROGRAMCI KATALOĞU**

Programı hazırlanması sürecinde, analiz ve tasarıma 2-3 gün,test ve raporlama işlemleri ise 2 gün sürmüştür.

**Ucus\_Numarasina\_Gore\_Ucus\_Listesi.cpp**

**#include <iostream>**

**#include <windows.h>**

**#include <mmsystem.h>**

**#include "Ucus.h"**

**#include "Ucus\_Numarasina\_Gore\_Ucus\_Listesi.h"**

**using namespace std;**

**Ucus\_Numarasina\_Gore\_Ucus\_Listesi::Ucus\_Numarasina\_Gore\_Ucus\_Listesi()**

**{**

**ilk=NULL;**

**}**

**void Ucus\_Numarasina\_Gore\_Ucus\_Listesi::ekle(Ucus \*yeni)**

**{**

**Ucus \*gecici,\*onceki;**

**if(ilk==NULL || yeni->ucusNoOku()<ilk->ucusNoOku()) //liste bossa veya en basa eklenecekse**

**{**

**yeni->sonrakiAyarla(ilk);**

**ilk=yeni;**

**}**

**else**

**{**

**onceki=ilk;**

**gecici=ilk->sonrakiOku();**

**while((gecici!=NULL) && (gecici->ucusNoOku()<yeni->ucusNoOku())) //eklenecek uygun yer araniyor**

**{**

**onceki=gecici;**

**gecici=gecici->sonrakiOku();**

**}**

**yeni->sonrakiAyarla(gecici); //onceki nesnesi ile gecici nesnesi arasina ekleniyor**

**onceki->sonrakiAyarla(yeni);**

**}**

**}**

**Ucus \*Ucus\_Numarasina\_Gore\_Ucus\_Listesi::ara(int ucusNo)**

**{**

**Ucus \*gecici=ilk;**

**while((gecici!=NULL) && (ucusNo>=gecici->ucusNoOku()))**

**{**

**if(gecici->ucusNoOku()==ucusNo) //bulunursa aranilan nesnenin adresi geriye donduruluyor**

**return gecici;**

**gecici=gecici->sonrakiOku();**

**}**

**return NULL; //bulunamazsa geriye NULL donduruluyor**

**}**

**void Ucus\_Numarasina\_Gore\_Ucus\_Listesi::cikar(Ucus \*cikarilan)**

**{**

**Ucus \*gecici,\*onceki=ilk;**

**if(cikarilan==ilk) //cikarilan listenin ilk nesnesi ise**

**{**

**ilk=ilk->sonrakiOku();**

**}**

**else**

**{**

**gecici=onceki->sonrakiOku();**

**while((gecici!=NULL) && (gecici!=cikarilan))**

**{**

**onceki=gecici;**

**gecici=gecici->sonrakiOku();**

**}**

**onceki->sonrakiAyarla(gecici->sonrakiOku());**

**}**

**}**

**void Ucus\_Numarasina\_Gore\_Ucus\_Listesi::luksListele()**

**{**

**Ucus \*gecici=ilk;**

**int sagSerbest=0;**

**while(gecici!=NULL)**

**{**

**if(gecici->luksKoltukOku()>0)**

**{**

**sagSerbest++;**

**cout<<\*gecici;**

**}**

**gecici=gecici->sonrakiOku();**

**}**

**if(sagSerbest==0)**

**{**

**cout<<"\n\t >>> Luks Koltuk Iceren Ucus Yok...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**}**

**ostream &operator<<(ostream &cikti,const Ucus\_Numarasina\_Gore\_Ucus\_Listesi &ucus) //<< operatoru asiri yukleniyor**

**{**

**Ucus \*gecici=ucus.ilk;**

**while(gecici!=NULL)**

**{**

**cikti<<\*gecici;**

**gecici=gecici->sonrakiOku();**

**}**

**return cikti;**

**}**

**Ucus\_Numarasina\_Gore\_Ucus\_Listesi.h**

**class Ucus\_Numarasina\_Gore\_Ucus\_Listesi {**

**friend ostream &operator<<(ostream &,const Ucus\_Numarasina\_Gore\_Ucus\_Listesi &);**

**private:**

**Ucus \*ilk;**

**public:**

**Ucus\_Numarasina\_Gore\_Ucus\_Listesi();**

**void ekle(Ucus \*);**

**Ucus \*ara(int);**

**void cikar(Ucus \*);**

**void luksListele();**

**};**

**Kalkis\_Yerine\_Gore\_Ucus\_Listesi.cpp**

**#include <iostream>**

**#include <windows.h>**

**#include <mmsystem.h>**

**#include "String.h"**

**#include "Ucus.h"**

**#include "Kalkis\_Yerine\_Gore\_Ucus\_Listesi.h"**

**using namespace std;**

**Kalkis\_Yerine\_Gore\_Ucus\_Listesi::Kalkis\_Yerine\_Gore\_Ucus\_Listesi() : listeBasi(0,0,0,0,0,0)**

**{**

**listeBasi.oncekiAyarla(&listeBasi);**

**listeBasi.cSonrakiAyarla(&listeBasi);**

**}**

**void Kalkis\_Yerine\_Gore\_Ucus\_Listesi::ekle(Ucus \*yeni)**

**{**

**Ucus \*gecici=listeBasi.cSonrakiOku();**

**while((gecici!=&listeBasi) && ((yeni->kalkisYeriOku())>(gecici->kalkisYeriOku()))) //alfabetik siraya gore uygun yer araniyor**

**gecici=gecici->cSonrakiOku();**

**while(((yeni->kalkisYeriOku())==(gecici->kalkisYeriOku())) && ((yeni->zamanOku()>(gecici->zamanOku())))) //kalkis zamani icin uygun yer araniyor**

**gecici=gecici->cSonrakiOku();**

**yeni->cSonrakiAyarla(gecici);**

**yeni->oncekiAyarla(gecici->oncekiOku());**

**(gecici->oncekiOku())->cSonrakiAyarla(yeni);**

**gecici->oncekiAyarla(yeni);**

**}**

**Ucus \*Kalkis\_Yerine\_Gore\_Ucus\_Listesi::ara(int ucusNo)**

**{**

**Ucus \*gecici=listeBasi.cSonrakiOku();**

**while(gecici!=&listeBasi)**

**{**

**if(gecici->ucusNoOku()==ucusNo) //bulunursa aranilan nesnenin adresi geriye donduruluyor**

**return gecici;**

**gecici=gecici->cSonrakiOku();**

**}**

**return NULL; //bulunamazsa geriye NULL donduruluyor**

**}**

**void Kalkis\_Yerine\_Gore\_Ucus\_Listesi::ara(String aranan,int s)**

**{**

**Ucus \*gecici=listeBasi.cSonrakiOku();**

**int sagSerbest=0;**

**switch(s)**

**{**

**case 0: //kalkis yerine gore arama**

**while((gecici!=&listeBasi) && (aranan>=(gecici->kalkisYeriOku())))**

**{**

**if((gecici->kalkisYeriOku())==aranan)**

**{**

**sagSerbest++;**

**cout<<\*gecici;**

**}**

**gecici=gecici->cSonrakiOku();**

**}**

**if(sagSerbest==0)**

**{**

**cout<<"\n\t >>> "<<aranan<<" Sehrinden Kalkan Bir Ucus Yok...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**break;**

**case 1: //varis yerine gore arama**

**while(gecici!=&listeBasi)**

**{**

**if((gecici->varisYeriOku())==aranan)**

**{**

**sagSerbest++;**

**cout<<\*gecici;**

**}**

**gecici=gecici->cSonrakiOku();**

**}**

**if(sagSerbest==0)**

**{**

**cout<<"\n\t >>> "<<aranan<<" Sehrine Olan Bir Ucus Yok...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**break;**

**}**

**}**

**void Kalkis\_Yerine\_Gore\_Ucus\_Listesi::ara(String kalkis,String varis)**

**{**

**Ucus \*gecici=listeBasi.oncekiOku();**

**int sagSerbest=0;**

**while((gecici!=&listeBasi) && ((gecici->kalkisYeriOku())>=kalkis))**

**{**

**if((gecici->kalkisYeriOku())==kalkis && (gecici->varisYeriOku())==varis)**

**{**

**sagSerbest++;**

**cout<<\*gecici;**

**}**

**gecici=gecici->oncekiOku();**

**}**

**if(sagSerbest==0)**

**{**

**cout<<"\n\t >>> "<<kalkis<<" Sehrinden "<<varis<<" Sehrine Olan Bir Ucus Yok...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**}**

**void Kalkis\_Yerine\_Gore\_Ucus\_Listesi::cikar(Ucus \*cikarilan)**

**{**

**(cikarilan->oncekiOku())->cSonrakiAyarla(cikarilan->cSonrakiOku());**

**(cikarilan->cSonrakiOku())->oncekiAyarla(cikarilan->oncekiOku());**

**}**

**ostream &operator<<(ostream &cikti,Kalkis\_Yerine\_Gore\_Ucus\_Listesi &ucus) //<< operatoru asiri yukleniyor**

**{**

**Ucus \*gecici=(ucus.listeBasi).cSonrakiOku();**

**while(gecici!=&(ucus.listeBasi))**

**{**

**cikti<<\*gecici;**

**gecici=gecici->cSonrakiOku();**

**}**

**return cikti;**

**}**

**Kalkis\_Yerine\_Gore\_Ucus\_Listesi.h**

**class Kalkis\_Yerine\_Gore\_Ucus\_Listesi {**

**friend ostream &operator<<(ostream &,Kalkis\_Yerine\_Gore\_Ucus\_Listesi &);**

**private:**

**Ucus listeBasi;**

**public:**

**Kalkis\_Yerine\_Gore\_Ucus\_Listesi();**

**void ekle(Ucus \*);**

**Ucus \*ara(int);**

**void ara(String,int);**

**void ara(String,String);**

**void cikar(Ucus \*);**

**};**

**Ucus.cpp**

**#include <iostream>**

**#include <iomanip>**

**#include "String.h"**

**#include "Ucus.h"**

**using namespace std;**

**Ucus::Ucus(int no,String kalkis,String varis,String zaman,int normal,int luks,Ucus \*onceki,Ucus \*sonraki,Ucus \*cSonraki)**

**: ucusNo(no),kalkisYeri(kalkis),varisYeri(varis),zaman(zaman),normalKoltukSayisi(normal)**

**{**

**luksKoltukAyarla(luks);**

**}**

**int Ucus::ucusNoOku()**

**{**

**return ucusNo;**

**}**

**String Ucus::kalkisYeriOku()**

**{**

**return kalkisYeri;**

**}**

**String Ucus::varisYeriOku()**

**{**

**return varisYeri;**

**}**

**String Ucus::zamanOku()**

**{**

**return zaman;**

**}**

**int Ucus::normalKoltukOku()**

**{**

**return normalKoltukSayisi;**

**}**

**int Ucus::luksKoltukOku()**

**{**

**return luksKoltukSayisi;**

**}**

**void Ucus::luksKoltukAyarla(int l)**

**{**

**luksKoltukSayisi=l;**

**}**

**Ucus \*Ucus::oncekiOku()**

**{**

**return onceki;**

**}**

**void Ucus::oncekiAyarla(Ucus \*o)**

**{**

**onceki=o;**

**}**

**Ucus \*Ucus::sonrakiOku()**

**{**

**return sonraki;**

**}**

**void Ucus::sonrakiAyarla(Ucus \*s)**

**{**

**sonraki=s;**

**}**

**Ucus \*Ucus::cSonrakiOku()**

**{**

**return cSonraki;**

**}**

**void Ucus::cSonrakiAyarla(Ucus \*s)**

**{**

**cSonraki=s;**

**}**

**ostream &operator<<(ostream &cikti,Ucus &ucus) //<< operatoru asiri yukleniyor**

**{**

**cikti<<" "<<left<<setw(7)<<ucus.ucusNoOku()<<" "**

**<<left<<setw(11)<<ucus.kalkisYeriOku()<<" "**

**<<left<<setw(10)<<ucus.varisYeriOku()<<" "**

**<<right<<setw(5)<<ucus.zamanOku()<<" "**

**<<right<<setw(13)<<ucus.normalKoltukOku()<<" "**

**<<right<<setw(11)<<ucus.luksKoltukOku()<<endl;**

**return cikti;**

**}**

**Ucus.h**

**class Ucus {**

**friend ostream &operator<<(ostream &,Ucus &);**

**private:**

**const int ucusNo;**

**const String kalkisYeri;**

**const String varisYeri;**

**const String zaman;**

**const int normalKoltukSayisi;**

**int luksKoltukSayisi;**

**Ucus \*onceki; //cift bagli liste icin onceki**

**Ucus \*sonraki; //tek bagli liste icin sonraki**

**Ucus \*cSonraki; //cift bagli liste icin sonraki**

**public:**

**Ucus(const int,const String,const String,const String,const int,int,Ucus \* = NULL,Ucus \* = NULL,Ucus \* = NULL);**

**int ucusNoOku();**

**String kalkisYeriOku();**

**String varisYeriOku();**

**String zamanOku();**

**int normalKoltukOku();**

**int luksKoltukOku();**

**void luksKoltukAyarla(int);**

**Ucus \*oncekiOku();**

**void oncekiAyarla(Ucus \*);**

**Ucus \*sonrakiOku();**

**void sonrakiAyarla(Ucus \*);**

**Ucus \*cSonrakiOku();**

**void cSonrakiAyarla(Ucus \*);**

**};**

**String.cpp**

**#include <iostream>**

**#include <iomanip>**

**#include <cstdio>**

**#include <cstring>**

**#include "String.h"**

**using namespace std;**

**String::String(int b)**

**{**

**boyut=b;**

**ptr=new char[boyut];**

**for(int i=0;i<boyut;i++)**

**ptr[i]='\0';**

**}**

**bool String::operator==(const String &str) //== operatoru asiri yukleniyor**

**{**

**if(strcmp(ptr,str.ptr))**

**return false;**

**else**

**return true;**

**}**

**bool String::operator>(const String &str) //> operatoru asiri yukleniyor**

**{**

**if(strcmp(ptr,str.ptr)>0)**

**return true;**

**else**

**return false;**

**}**

**bool String::operator>=(const String &str) //>= operatoru asiri yukleniyor**

**{**

**if(strcmp(ptr,str.ptr)>=0)**

**return true;**

**else**

**return false;**

**}**

**ostream &operator<<(ostream &cikti,const String &str) //<< operatoru asiri yukleniyor**

**{**

**int i;**

**char s[21];**

**for(i=0;i<str.boyut;i++)**

**{**

**if(str.ptr[i]!='\0')**

**s[i]=str.ptr[i];**

**else**

**break;**

**}**

**s[i]='\0';**

**cikti<<s;**

**return cikti;**

**}**

**istream &operator>>(istream &girdi,String &str) //>> operatoru asiri yukleniyor**

**{**

**char c;**

**for(int i=0;i<str.boyut;i++)**

**{**

**c=getchar();**

**if(c=='\n')**

**break;**

**else**

**if(i==0) //ilk harf buyuk harfe cevriliyor**

**c=toupper(c);**

**else //diger harfler kucuk harfe cevriliyor**

**c=tolower(c);**

**str.ptr[i]=c;**

**}**

**return girdi;**

**}**

**String.h**

**#include <iostream>**

**using namespace std;**

**class String {**

**friend ostream &operator<<(ostream &,const String &);**

**friend istream &operator>>(istream &,String &);**

**private:**

**int boyut;**

**char \*ptr;**

**public:**

**String(int=21);**

**bool operator==(const String &);**

**bool operator>(const String &);**

**bool operator>=(const String &);**

**};**

**main.cpp**

**#include <iostream>**

**#include <iomanip>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <windows.h>**

**#include <mmsystem.h>**

**#include "String.h"**

**#include "Ucus.h"**

**#include "Ucus\_Numarasina\_Gore\_Ucus\_Listesi.h"**

**#include "Kalkis\_Yerine\_Gore\_Ucus\_Listesi.h"**

**using namespace std;**

**int menuSecim();**

**void baslikYazdir();**

**void hakkinda();**

**void beklet();**

**int main()**

**{**

**system("title Havayolu Ucus Takip Sistemi");**

**system("color f9");**

**Ucus \*ucus;**

**Ucus\_Numarasina\_Gore\_Ucus\_Listesi liste1;**

**Kalkis\_Yerine\_Gore\_Ucus\_Listesi liste2;**

**String \*kalkisYeri,\*varisYeri,\*kalkisZamani;**

**int ucusNo,normalKoltukSayisi,luksKoltukSayisi;**

**int secim;**

**do{**

**secim=menuSecim();**

**switch(secim)**

**{**

**case 1:**

**cout<<"\n\t >> Lutfen Ucus Numarasini Giriniz : ";**

**cin>>ucusNo;**

**if(liste1.ara(ucusNo)==NULL)**

**{**

**kalkisYeri=new String();**

**varisYeri=new String();**

**kalkisZamani=new String();**

**cout<<"\n\t >> Kalkis Yerini Giriniz : ";**

**fflush(stdin);**

**cin>>\*kalkisYeri;**

**cout<<"\n\t >> Varis Yerini Giriniz : ";**

**fflush(stdin);**

**cin>>\*varisYeri;**

**cout<<"\n\t >> Kalkis Zamanini Giriniz : ";**

**fflush(stdin);**

**cin>>\*kalkisZamani;**

**cout<<"\n\t >> Normal Koltuk Sayisini Giriniz : ";**

**cin>>normalKoltukSayisi;**

**luksKoltukSayisi=0;**

**ucus=new Ucus(ucusNo,\*kalkisYeri,\*varisYeri,\*kalkisZamani,normalKoltukSayisi,luksKoltukSayisi);**

**liste1.ekle(ucus);**

**liste2.ekle(ucus);**

**cout<<"\n\t \*\*\* Ucusunuz Basariyla Eklenmistir..."<<endl;**

**}**

**else**

**{**

**cout<<"\n\t >>> Bu Numaraya Sahip Ucus Daha Onceden Eklenmistir...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**break;**

**case 2:**

**cout<<"\n\t >> Lutfen Ucus Numarasini Giriniz : ";**

**cin>>ucusNo;**

**if(liste1.ara(ucusNo)==NULL)**

**{**

**kalkisYeri=new String();**

**varisYeri=new String();**

**kalkisZamani=new String();**

**cout<<"\n\t >> Kalkis Yerini Giriniz : ";**

**fflush(stdin);**

**cin>>\*kalkisYeri;**

**cout<<"\n\t >> Varis Yerini Giriniz : ";**

**fflush(stdin);**

**cin>>\*varisYeri;**

**cout<<"\n\t >> Kalkis Zamanini Giriniz : ";**

**fflush(stdin);**

**cin>>\*kalkisZamani;**

**cout<<"\n\t >> Normal Koltuk Sayisini Giriniz : ";**

**cin>>normalKoltukSayisi;**

**cout<<"\n\t >> Luks Koltuk Sayisini Giriniz : ";**

**cin>>luksKoltukSayisi;**

**ucus=new Ucus(ucusNo,\*kalkisYeri,\*varisYeri,\*kalkisZamani,normalKoltukSayisi,luksKoltukSayisi);**

**liste1.ekle(ucus);**

**liste2.ekle(ucus);**

**cout<<"\n\t \*\*\* Ucusunuz Basariyla Eklenmistir..."<<endl;**

**}**

**else**

**{**

**cout<<"\n\t >>> Bu Numaraya Sahip Ucus Daha Onceden Eklenmistir...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**break;**

**case 3:**

**cout<<"\n\t >> Lutfen Ucus Numarasini Giriniz : ";**

**cin>>ucusNo;**

**if((ucus=liste1.ara(ucusNo))==NULL)**

**{**

**cout<<"\n\t >>> Malesef Bu Numarali Bir Ucus Bulunmamaktadir...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**else**

**{**

**cout<<"\n\t >> Lutfen Yeni Luks Koltuk Sayisini Giriniz : ";**

**cin>>luksKoltukSayisi;**

**ucus->luksKoltukAyarla(luksKoltukSayisi);**

**cout<<"\n\t \*\*\* Koltuk Sayisi Basariyla Guncellenmistir..."<<endl;**

**}**

**break;**

**case 4:**

**cout<<"\n\t >> Lutfen Ucus Numarasini Giriniz : ";**

**cin>>ucusNo;**

**if((ucus=liste1.ara(ucusNo))==NULL)**

**{**

**cout<<"\n\t >>> Malesef Bu Numarali Bir Ucus Bulunmamaktadir...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**else**

**{**

**liste1.cikar(ucus);**

**liste2.cikar(ucus);**

**delete ucus;**

**cout<<"\n\t \*\*\* Ucus Basariyla Iptal edilmistir..."<<endl;**

**}**

**break;**

**case 5:**

**cout<<"\n\t >> Lutfen Ucus Numarasini Giriniz : ";**

**cin>>ucusNo;**

**if((ucus=liste1.ara(ucusNo))==NULL)**

**{**

**cout<<"\n\t >>> Malesef Bu Numarali Bir Ucus Bulunmamaktadir...!"<<endl;**

**PlaySound(TEXT("SystemHand"),NULL,SND\_ALIAS);**

**}**

**else**

**{**

**baslikYazdir();**

**cout<<\*ucus;**

**}**

**break;**

**case 6:**

**baslikYazdir();**

**liste1.luksListele();**

**break;**

**case 7:**

**varisYeri=new String();**

**cout<<"\n\t >> Lutfen Varis Yerini Giriniz : ";**

**fflush(stdin);**

**cin>>\*varisYeri;**

**baslikYazdir();**

**liste2.ara(\*varisYeri,1);**

**delete varisYeri;**

**break;**

**case 8:**

**kalkisYeri=new String();**

**cout<<"\n\t >> Lutfen Kalkis Yerini Giriniz : ";**

**fflush(stdin);**

**cin>>\*kalkisYeri;**

**baslikYazdir();**

**liste2.ara(\*kalkisYeri,0);**

**delete kalkisYeri;**

**break;**

**case 9:**

**kalkisYeri=new String();**

**varisYeri=new String();**

**cout<<"\n\t >> Lutfen Kalkis Yerini Giriniz : ";**

**fflush(stdin);**

**cin>>\*kalkisYeri;**

**cout<<"\n\t >> Lutfen Varis Yerini Giriniz : ";**

**fflush(stdin);**

**cin>>\*varisYeri;**

**baslikYazdir();**

**liste2.ara(\*kalkisYeri,\*varisYeri);**

**delete kalkisYeri;**

**delete varisYeri;**

**break;**

**case 10:**

**baslikYazdir();**

**cout<<liste1;**

**cout<<endl;**

**baslikYazdir();**

**cout<<liste2;**

**break;**

**case 11:**

**hakkinda();**

**}**

**}while(secim!=11);**

**return 0;**

**}**

**int menuSecim()**

**{**

**int secim;**

**cout<<endl**

**<<"\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MENU \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl**

**<<"\n\t 1. Luks koltuk icermeyen yeni bir ucusun eklenmesi"<<endl**

**<<"\t 2. Luks koltuk iceren yeni bir ucusun eklenmesi"<<endl**

**<<"\t 3. Bir ucusun luks koltuk sayisinin guncellenmesi"<<endl**

**<<"\t 4. Bir ucusun iptal edilmesi"<<endl**

**<<"\t 5. Bir ucusun bilgilerinin goruntulenmesi"<<endl**

**<<"\t 6. Luks koltuk iceren ucuslarin listelenmesi"<<endl**

**<<"\t 7. Bir yere olan ucuslarin listelenmesi"<<endl**

**<<"\t 8. Bir yerden kalkan ucuslarin listelenmesi"<<endl**

**<<"\t 9. Bir yerden bir yere olan ucuslarin listelenmesi"<<endl**

**<<"\t10. Tum ucuslarin listelenmesi"<<endl**

**<<"\t11. Cikis"<<endl**

**<<endl;**

**do{**

**cout<<"\n\t >>> Lutfen seciminizi yapiniz : ";**

**cin>>secim;**

**}while(secim<1 || secim>11);**

**return secim;**

**}**

**void baslikYazdir()**

**{**

**cout<<endl**

**<<"\n Ucus No"<<" "<<"Kalkis Yeri"<<" "<<"Varis Yeri"<<" "**

**<<"Zaman"<<" "<<"Normal Koltuk"<<" "<<"Luks Koltuk"<<endl**

**<<" -------"<<" "<<"-----------"<<" "<<"----------"<<" "**

**<<"-----"<<" "<<"-------------"<<" "<<"-----------"<<endl;**

**}**

**void hakkinda()**

**{**

**for(int i=15;i>0;i--)**

**{**

**system("cls");**

**for(int j=0;j<i;j++)**

**cout<<endl;**

**cout<<"\n\t \*\*\*...Programimizi Kullandiginiz Icin Tesekkur Ederiz...\*\*\*\n\n"<<endl**

**<<"\t Yapimcilar"<<endl**

**<<"\t ----------\n"<<endl**

**<<"\t Onur Cem Senel\n"<<endl**

**<<"\t Furkan Tanriverdi\n"<<endl**

**<<"\t Burak Yabar\n"<<endl;**

**beklet();**

**}**

**PlaySound(TEXT("SystemExit"),NULL,SND\_ALIAS);**

**}**

**void beklet()**

**{**

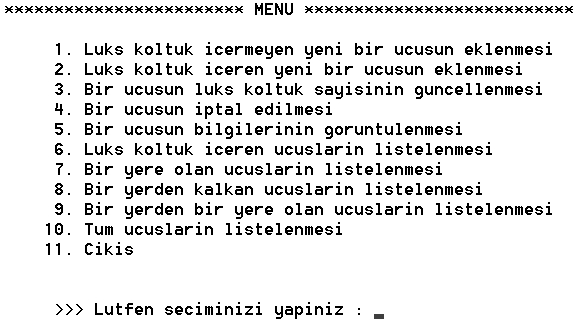
**for(long int i=0;i<50000000;i++) {}**

**}**

**KULLANICI KATALOĞU**

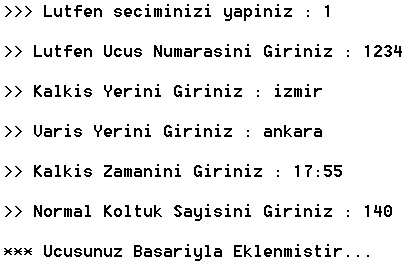
**1 )** Hava yolu bilet satışı programını başlattığınız zaman ekrana aşağıdaki gibi bir menü gelecektir.Buradan yapmak istediğiniz seçime ilişkin seçeneği seçebilirsiniz. (Bakınız RESİM 1)

**RESİM 1**



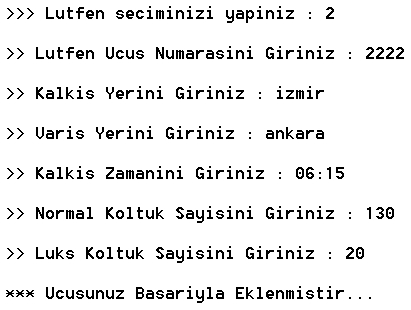
**2)** Programı kullanarak lüks koltuk içermeyen bir uçuş eklemek istiyorsanız 1 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 2)

**RESİM 2**



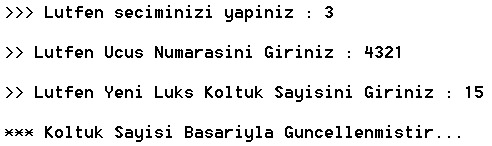
**3)** Programı kullanarak lüks koltuk içeren bir uçuş eklemek istiyorsanız 2 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 3)

**RESİM 3**



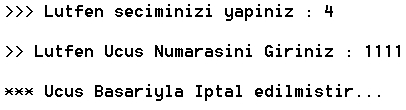
**4)** Programı kullanarak lüks koltuk içeren bir uçuşun lüks koltuk sayısını güncellemek istiyorsanız 3 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 4)

**RESİM 4**



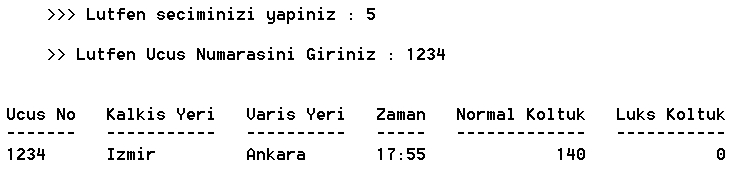
**5)** Programı kullanarak bir uçuşu iptal etmek istiyorsanız 4 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 5)

**RESİM 5**



**6)** Programı kullanarak bir uçuşun bilgilerini listelemek istiyorsanız 5 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 6)

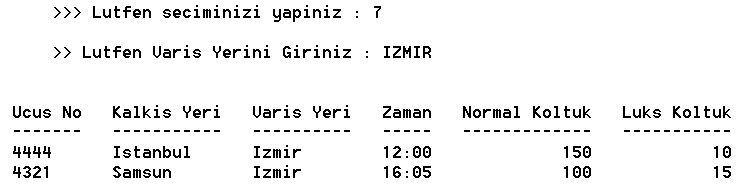
**RESİM 6**



**7)** Programı kullanarak lüks koltuk içeren uçuşları listelemek istiyorsanız 6 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 7)

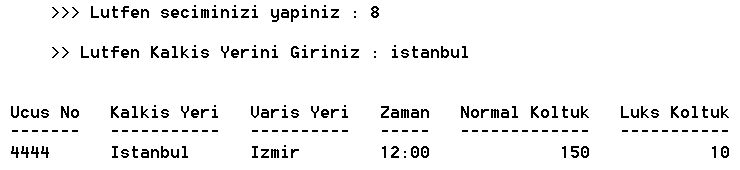
**RESİM 7**

**8)** Programı kullanarak varış yerine göre uçuşları listelemek istiyorsanız 7 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 8)

**RESİM 8**

**9)** Programı kullanarak kalkış yerine göre uçuşları listelemek istiyorsanız 8 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 9)

**RESİM 9**



**10)** Programı kullanarak kalkış yerini ve varış yerini girerek uygun uçuşları listelemek istiyorsanız 9 nolu seçeneği seçip sizden istenilenleri doğru bir şekilde girerek işleminizi başarıyla gerçekleştirebilirsiniz. (Bakınız RESİM 10)

**RESİM 10**

