**07.05.2023**

logo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**ERZURUM ATATÜRK ÜNİVERSİTESİ**

**YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**NESNEYE YÖNELİK PROGRAMALAMA**

**PROJE-2 HAVAYOLU BİLET SATIŞ SİSTEMİ-2**

**RAPORU HAZIRLAYAN: BEYZANUR TOYSÖZ**

**ÖĞRENCİ NUMARASI: 220710011**

**RAPORU HAZIRLAYAN: NİSA KOPUZ**

**ÖĞRENCİ NUMARASI: 220710015**

**RAPORU HAZIRLAYAN: HASRET İŞLER**

**ÖĞRENCİ NUMARASI: 220710017**

**İÇİNDEKİLER**

**Analiz………………………………………………………3**

**Tasarım…………………………………………………….4**

**Programcı Kataloğu………………………………………7**

**Kullanıcı Kataloğu………………………………………..34**

**ANALİZ**

Bir havayolu şirketi tarafından kendilerine ait günlük uçuşların ve bu uçuşlara ilişkin yapılan bilet satışlarının takip edilmesini sağlayacak bir uygulama geliştirilmesi istenmektedir. Geliştirilmesi istenen bu program yalnızca kullanıcının değil havayolu şirketi bünyesinde yer alan çalışanların da rahatlıkla kullanabilecekleri ve yaşanan bir aksilikte uçuş numaralarını silerek uçuşları iptal ettirebilecekleri ya da uçağın kalkış zamanını güncelleyebilecekleri bir imkan sunmaktadır. Kullanıcıdan alınan uçuş numarası ile bu uçuş numarasına ait bir uçuşun bulunup bulunmadığından haberdar edilmesi ise programın bir diğer amacıdır. Her veri girişinde kontrol amaçlı olarak uçuşların boş koltuk sayısına bakılarak doluluk oranları hesaplanmakta ve bilet satın alınmadan önce istenilen koltuk sayısına göre bilet fiyatları hakkında bilgilendirme yapılmaktadır. Kalkış ve varış yerine bağlı olarak değişen başlangıç bilet fiyatları aritmetik bir hesaplamanın eklenmesi sonucu gerçek bilet fiyatlarına erişmelerini sağlamaktadır. Bizden geliştirilmesi istenilen ve içerisinde birçok farklı içerik barındıran “Havayolu Bilet Satış Sistemi” programı bütün bu özelliklerin yanı sıra kullanıcılara o zamana kadar satın aldıkları bütün biletlerin listelenmesini sağlayıp ek olarak diledikleri takdirde ortalama bilet fiyatı gibi bilgilere erişme imkanını da sunmaktadır.

**TASARIM**

Aşağıda program için geliştirilen fonksiyonların ve veri yapılarının prototipleri ve işlevleri verilmiştir.

**string BuyukHarf(string &s);**

Programda kullanılan string verilerin büyük harfe dönüştürülmesini sağlayarak hem yazımlarını hem de karşılaştırılmalarını kolaylaştırır.

**int NumaraVarMi(int numara);**

Programda kullanılan fonksiyonların hata mesajlarını kolaylaştırmak için listede girilen uçuş numarası olup olmadığını kontrol eder. Varsa geriye 1, yoksa geriye 0 döndürür.

**int KalkisVarMi(int numara);**

Programda kullanılan fonksiyonların hata mesajlarını kolaylaştırmak için listede girilen kalkış yeri olup olmadığını kontrol eder. Varsa geriye 1, yoksa geriye 0 döndürür.

**int VarisVarMi(int numara);**

Programda kullanılan fonksiyonların hata mesajlarını kolaylaştırmak için listede girilen varış yeri olup olmadığını kontrol eder. Varsa geriye 1, yoksa geriye 0 döndürür.

**void UcusEkle(int numara**,**string kalkisYeri, string varisYeri, int kalkisSaati, int kalkisDakikasi, int koltukKapasitesi, float biletFiyati);**

Listeye yeni bir uçuş ekler. Eğer alınan uçuş numarası başka bir uçuşa aitse hata verecek ve uçuş ekleme işlemini gerçekleştirmeyecektir.

**void ZamanGuncelle(int numara);**

Listede istenilen uçuşun kalkış zamanını değiştirir. Eğer girilen uçuş numarasına sahip uçuş yoksa hata verecek ve zaman değiştirme işlemini gerçekleştirmeyecektir.

**void UcusSil(int numara);**

Listeden istenilen bir uçuşu siler. Eğer girilen uçuş numarasına sahip bir uçuş zaten yoksa bu durumu kullanıcıya bildirip işlem gerçekleştirmeyecektir.

**void UcusBilgisi(int numara);**

Listeden istenilen bir uçuşa ait bilgileri sıralar. Eğer girilen uçuş numarasına sahip bir uçuş yoksa hata verecek ve işlem gerçekleştirmeyecektir.

**void YuzdeElliAlti();**

Listedeki doluluk oranı %50 altı olan uçuşları sıralar.

**void KalkisYerineGore(string kalkisYeri);**

Listeden istenilen yerden kalkan uçuşları bulur ve sıralar. Eğer girilen kalkış yerine sahip bir uçuş yoksa hata verecek ve işlem gerçekleştirmeyecektir.

**void KalkisveVarisYerineGore(string kalkisYeri);**

Listeden istenilen yerden kalkan ve istenilen yerde inecek olan uçuşları bulur ve sıralar. Eğer girilen kalkış ve varış yerine sahip bir uçuş yoksa hata verecek ve işlem gerçekleştirmeyecektir.

**void BiletSatinAl(int numara);**

Listedeki istenilen bir uçuşa ilişkin bilet satın alınmasını sağlar. Eğer alınan uçuş numarasına ilişkin uçuş yoksa hata verecek ve işlem gerçekleştirmeyecektir. Yeterli sayıda boş koltuk varsa bir seferde en fazla 10 bilet almak koşuluyla satın alım işlemi gerçekleştirilebilir.

**void BiletIptali(int numara);**

Bilet listesinden satılmış bir bileti iptal eder. Eğer zaten girilen kıstaslara uygun bir bilet yoksa hata verecek ve işlem gerçekleştirmeyecektir.

**void BiletBilgisi(int numara);**

Listeden girilen uçuşa ilişkin biletlere ait bilgileri sıralar. Eğer bilet yoksa hata verecek ve işlem gerçekleştirmeyecektir.

**void YolcuBilgisi(string tcno);**

Listeden girilen TC numarasına ait biletleri sıralar. Eğer bilet yoksa hata verecek ve işlem gerçekleştirmeyecektir.

**class UcusListesi{};**

Struct yapıcıyla oluşturulmuş uçuşları liste halinde saklayacak fonksiyonları içerir.

**class BiletListesi{};**

Struct yapıcıyla oluşturulmuş biletleri liste halinde saklayacak fonksiyonları içerir.

**PROGRAMCI KATALOĞU**

|  |  |
| --- | --- |
| **Analiz** | **1 gün** |
| **Tasarım** | **4 gün** |
| **Gerçekleştirim** | **9 gün** |
| **Test** | **2 gün** |
| **Raporlama** | **1 gün** |
| **Toplam** | **17 gün** |

* Projenin her aşaması birlikte yürütülmüştür.

#include <iostream>

#include <iomanip>

#include <algorithm>

#include <ctype.h>

#include <cstring>

using namespace std;

struct bilgi{

int numara;

string kalkisYeri;

string varisYeri;

int kalkisSaati;

int kalkisDakikasi;

int koltukKapasitesi;

int bosKoltukSayisi;

float biletFiyati;

bilgi\* sonraki;

};

struct bilet{

int numara;

string TCno;

float odenenBilet;

bilet\* next;

};

class UcusListe {

public:

UcusListe()

{

ilk = NULL;

}

string BuyukHarf(string &s)

{

for(int i=0;i<s.length();i++)

{

s[i]=toupper(s[i]);

}

return s;

}

int NumaraVarMi(int numara)

{

int sayi=0;

bilgi \*p=NULL;

p=ilk;

while(p!=NULL)

{

if(p->numara==numara)

{

sayi++;

break;

}

p=p->sonraki;

}

if(sayi==0)

{

return 0;

}

else

{

return 1;

}

}

int KalkisVarMi(string kalkisYeri)

{

int sayi=0;

bilgi \*p=NULL;

p=ilk;

while(p!=NULL)

{

if(BuyukHarf(kalkisYeri)==BuyukHarf(p->kalkisYeri))

{

sayi++;

break;

}

p=p->sonraki;

}

if(sayi==0)

{

return 0;

}

else

{

return 1;

}

}

int VarisVarMi(string varisYeri)

{

int sayi=0;

bilgi \*p=NULL;

p=ilk;

while(p!=NULL)

{

if(BuyukHarf(varisYeri)==BuyukHarf(p->varisYeri))

{

sayi++;

break;

}

p=p->sonraki;

}

if(sayi==0)

{

return 0;

}

else

{

return 1;

}

}

void UcusEkle(int numara, string kalkisYeri, string varisYeri, int kalkisSaati, int kalkisDakikasi, int koltukKapasitesi, float biletFiyati)

{

bilgi \*yeni=new bilgi;

if(NumaraVarMi(numara))

{

cout<<"\nGirdiginiz numara baska bir ucus tarafindan kullanilmaktadir.\n";

}

else

{

cout<<"Kalkis yeri giriniz:";

cin>>kalkisYeri;

cout<<"Varis yeri giriniz:";

cin>>varisYeri;

cout<<"Kalkis saatini giriniz:";

cin>>kalkisSaati;

cout<<"Kalkis dakikasini giriniz:";

cin>>kalkisDakikasi;

cout<<"Koltuk kapasitesini giriniz:";

cin>>koltukKapasitesi;

cout<<"Baslangic bilet fiyatini giriniz:";

cin>>biletFiyati;

yeni->numara = numara;

yeni->kalkisYeri = BuyukHarf(kalkisYeri);

yeni->varisYeri = BuyukHarf(varisYeri);

yeni->kalkisSaati = kalkisSaati;

yeni->kalkisDakikasi = kalkisDakikasi;

yeni->koltukKapasitesi = koltukKapasitesi;

yeni->bosKoltukSayisi = koltukKapasitesi;

yeni->biletFiyati = biletFiyati;

yeni->sonraki = ilk;

ilk = yeni;

cout<<"\nYeni ucusa iliskin bilgiler su sekildedir:\n\n"<<"Ucus No \tKalkis Yeri Varis Yeri \tZamani \tKapasite \tBilet Fiyati\n------- \t----------- ---------- \t------ \t---------- \t------------\n";

cout<<yeni->numara<<"\t\t"<<yeni->kalkisYeri<<"\t\t"<<yeni->varisYeri<<"\t\t"<<setfill('0')<<setw(2)<<yeni->kalkisSaati<<":"<<setfill('0')<<setw(2)<<yeni->kalkisDakikasi<<"\t\t"<<yeni->koltukKapasitesi<<"\t\t"<<yeni->biletFiyati<<"\n";

cout<<"\nUcusunuz basariyla eklenmistir.\n";

}

}

void ZamanGuncelle(int numara)

{

bilgi\* simdiki=ilk;

int guncelKalkisSaati,guncelKalkisDakikasi;

float doluluk;

if(NumaraVarMi(numara))

{

while(simdiki!=NULL)

{

if(simdiki->numara==numara)

{

doluluk=static\_cast<float>((simdiki->koltukKapasitesi)-(simdiki->bosKoltukSayisi))\*100/(simdiki->koltukKapasitesi);

cout<<"\n\nUcus numarasina ait bilgiler su sekildedir: "<<endl<<"\n\nUcus No \tKalkis Yeri Varis Yeri \tZamani \tKapasite \tBos Koltuk \tDoluluk %%\n------- \t----------- ---------- \t------ \t-------- \t---------- \t---------\n";

cout<<simdiki->numara<<"\t\t"<<simdiki->kalkisYeri<<"\t\t"<<simdiki->varisYeri<<"\t\t"<<setfill('0')<<setw(2)<<simdiki->kalkisSaati<<":"<<setfill('0')<<setw(2)<<simdiki->kalkisDakikasi<<"\t\t"<<simdiki->koltukKapasitesi<<"\t\t"<<simdiki->bosKoltukSayisi<<"\t\t"<<doluluk;

cout<<"\n\nYeni kalkisa ait zaman bilgilerini giriniz: ";

cout<<"\n\nSaati: ";

cin>>guncelKalkisSaati;

cout<<"Dakikasi: ";

cin>>guncelKalkisDakikasi;

simdiki->kalkisSaati = guncelKalkisSaati;

simdiki->kalkisDakikasi = guncelKalkisDakikasi;

cout<<"\n\nUcus numarasina ait guncellenmis bilgiler su sekildedir: "<<endl<<"\n\nUcus No \tKalkis Yeri Varis Yeri \tZamani \tKapasite \tBos Koltuk \tDoluluk %%\n------- \t----------- ---------- \t------ \t-------- \t---------- \t---------\n";

cout<<simdiki->numara<<"\t\t"<<simdiki->kalkisYeri<<"\t\t"<<simdiki->varisYeri<<"\t\t"<<setfill('0')<<setw(2)<<simdiki->kalkisSaati<<":"<<setfill('0')<<setw(2)<<simdiki->kalkisDakikasi<<"\t\t"<<simdiki->koltukKapasitesi<<"\t\t"<<simdiki->bosKoltukSayisi<<"\t\t"<<doluluk;

break;

}

simdiki=simdiki->sonraki;

}

}

else

{

cout<<"\nAradiginiz ucusa iliskin kayit bulunmamaktadir.\n";

}

}

int BilgiCikar(int numara,string kalkisYeri, string varisYeri, int kalkisSaati, int kalkisDakikasi, int bosKoltukSayisi, float biletFiyati)

{

bilgi \*simdiki=ilk;

if(NumaraVarMi(numara))

{

while(simdiki!=NULL)

{

if(simdiki->numara==numara)

{

BuyukHarf(kalkisYeri)=simdiki->kalkisYeri;

BuyukHarf(varisYeri)=simdiki->varisYeri;

kalkisSaati=simdiki->kalkisSaati;

kalkisDakikasi=simdiki->kalkisDakikasi;

bosKoltukSayisi=simdiki->bosKoltukSayisi;

return 1;

}

simdiki=simdiki->sonraki;

}

}

else

{

cout<<"Aradiginiz ucusa ilişkin kayit bulunmamaktadir.";

return 0;

}

}

void KoltukGuncelle(int numara,int biletSayisi,int secenek)

{

bilgi\* simdiki=ilk;

int guncelKalkisSaati,guncelKalkisDakikasi;

float doluluk;

if(NumaraVarMi(numara))

{

while(simdiki!=NULL)

{

if(simdiki->numara==numara)

{

if(secenek==1)

{

simdiki->bosKoltukSayisi=(simdiki->bosKoltukSayisi)+biletSayisi;

}

else

{

simdiki->bosKoltukSayisi=(simdiki->bosKoltukSayisi)-biletSayisi;

}

}

simdiki=simdiki->sonraki;

}

}

else

{

cout<<"\nAradiginiz ucusa iliskin kayit bulunmamaktadir.\n";

}

}

void UcusSil(int numara)

{

if(NumaraVarMi(numara))

{

cout <<"\n"<<numara<< " numarali ucus silindi." << endl;

if (ilk->numara == numara)

{

bilgi\* silinecek = ilk;

ilk = ilk->sonraki;

delete silinecek;

return;

}

bilgi\* onceki = ilk;

bilgi\* simdiki = ilk->sonraki;

while (simdiki != NULL)

{

if (simdiki->numara == numara)

{

onceki->sonraki = simdiki->sonraki;

delete simdiki;

}

onceki = simdiki;

simdiki = simdiki->sonraki;

}

}

else

{

cout<<"\nSilmek istediginiz ucusa iliskin kayit bulunmamaktadir.\n";

}

}

void UcusBilgisi(int numara)

{

bilgi\* simdiki=ilk;

float doluluk;

if(NumaraVarMi(numara))

{

while(simdiki!=NULL)

{

if(simdiki->numara==numara)

{

doluluk=static\_cast<float>((simdiki->koltukKapasitesi)-(simdiki->bosKoltukSayisi))\*100/(simdiki->koltukKapasitesi);

cout<<"\n\nUcus No \tKalkis Yeri Varis Yeri \tZamani \tKapasite \tBos Koltuk \tDoluluk %%\n------- \t----------- ---------- \t------ \t-------- \t---------- \t---------\n";

cout<<simdiki->numara<<"\t\t"<<simdiki->kalkisYeri<<"\t\t"<<simdiki->varisYeri<<"\t\t"<<setfill('0')<<setw(2)<<simdiki->kalkisSaati<<":"<<setfill('0')<<setw(2)<<simdiki->kalkisDakikasi<<"\t\t"<<simdiki->koltukKapasitesi<<"\t\t"<<simdiki->bosKoltukSayisi<<"\t\t"<<doluluk;

break;

}

simdiki=simdiki->sonraki;

}

}

else

{

cout<<"\nAradiginiz ucusa iliskin kayit bulunmamaktadir.\n";

}

}

void YuzdeElliAlti()

{

bilgi\* elli=ilk;

int i=0;

float doluluk;

while(elli!=NULL)

{

doluluk=static\_cast<float>((elli->koltukKapasitesi)-(elli->bosKoltukSayisi))\*100/(elli->koltukKapasitesi);

if(doluluk<50)

{

i++;

if(i==1)

{

cout<<"\n\nUcus No \tKalkis Yeri Varis Yeri \tZamani \tKapasite \tBos Koltuk \tDoluluk %%\n------- \t----------- ---------- \t------ \t-------- \t---------- \t---------\n";

}

cout<<elli->numara<<"\t\t"<<elli->kalkisYeri<<"\t\t"<<elli->varisYeri<<"\t\t"<<setfill('0')<<setw(2)<<elli->kalkisSaati<<":"<<setfill('0')<<setw(2)<<elli->kalkisDakikasi<<"\t\t"<<elli->koltukKapasitesi<<"\t\t"<<elli->bosKoltukSayisi<<"\t\t"<<doluluk<<"\n";

}

elli=elli->sonraki;

}

}

void KalkisYerineGore(string kalkisYeri)

{

int i=0;

float doluluk;

bilgi\* simdiki = ilk;

cout<<"kalkis yerini giriniz:";

cin>>kalkisYeri;

if(KalkisVarMi(kalkisYeri))

{

while (simdiki != NULL)

{

if(BuyukHarf(kalkisYeri)==BuyukHarf(simdiki->kalkisYeri))

{

doluluk=static\_cast<float>((simdiki->koltukKapasitesi)-(simdiki->bosKoltukSayisi))\*100/(simdiki->koltukKapasitesi);

i++;

if(i==1)

{

cout<<"\n\nUcus No \tKalkis Yeri Varis Yeri \tZamani \tKapasite \tBos Koltuk \tDoluluk %%\n------- \t----------- ---------- \t------ \t-------- \t---------- \t---------\n";

}

cout<<simdiki->numara<<"\t\t"<<simdiki->kalkisYeri<<"\t\t"<<simdiki->varisYeri<<"\t\t"<<setfill('0')<<setw(2)<<simdiki->kalkisSaati<<":"<<setfill('0')<<setw(2)<<simdiki->kalkisDakikasi<<"\t\t"<<simdiki->koltukKapasitesi<<"\t\t"<<simdiki->bosKoltukSayisi<<"\t\t"<<doluluk;

simdiki = simdiki->sonraki;

}

}

}

else

{

cout<<"\nAradiginiz sehirden kalkan ucus bulunmamaktadir.\n";

}

}

void KalkisVeVarisYerineGore(string kalkisYeri, string varisYeri)

{

int i=0;

float doluluk;

bilgi\* simdiki = ilk;

cout<<"Kalkis yerini giriniz:";

cin>>kalkisYeri;

cout<<"Varis yerini giriniz:";

cin>>varisYeri;

if(KalkisVarMi(kalkisYeri)&&VarisVarMi(varisYeri))

{

while (simdiki != NULL)

{

if(BuyukHarf(kalkisYeri)==BuyukHarf(simdiki->kalkisYeri) && BuyukHarf(varisYeri)==BuyukHarf(simdiki->varisYeri))

{

doluluk=static\_cast<float>((simdiki->koltukKapasitesi)-(simdiki->bosKoltukSayisi))\*100/(simdiki->koltukKapasitesi);

i++;

if(i==1)

{

cout<<"\n\nUcus No \tKalkis Yeri Varis Yeri \tZamani \tKapasite \tBos Koltuk \tDoluluk %%\n------- \t----------- ---------- \t------ \t-------- \t---------- \t---------\n";

}

cout<<simdiki->numara<<"\t\t"<<simdiki->kalkisYeri<<"\t\t"<<simdiki->varisYeri<<"\t\t"<<setfill('0')<<setw(2)<<simdiki->kalkisSaati<<":"<<setfill('0')<<setw(2)<<simdiki->kalkisDakikasi<<"\t\t"<<simdiki->koltukKapasitesi<<"\t\t"<<simdiki->bosKoltukSayisi<<"\t\t"<<doluluk;

simdiki = simdiki->sonraki;

}

}

}

else

{

cout<<"\nAradiginiz ucus rotasina sahip ucus bulunmamaktadir.\n";

}

}

private:

bilgi\* ilk;

};

/\*\*

class BiletListe::public UcusListe{

public:

void BiletSatinAl(int numara)

{

int biletSayisi,a,i=100,j,bosKoltukOrani;;

char odemeKabul;

float toplamFiyat,k=0.9;

float ucret[9000];

string tcNo;

bilgi\* simdiki = ilk;

bilgi \*yeni = new bilgi;

if(NumaraVarMi(numara))

{

cout<<"\n\nUcus numarasina ait bilgiler su sekildedir: "<<endl;

cout<<"\n\nUcus No\t\tKalkis Yeri\t\tVaris Yeri\t\tZamani\t\tKapasite\t\tBilet Fiyati\n-------\t\t-----------\t\t----------\t\t------\t\t----------\t\t------------\n";

cout<<simdiki->numara<<"\t\t";

cout<<simdiki->kalkisYeri<<"\t\t\t";

cout<<simdiki->varisYeri<<"\t\t\t";

cout<<simdiki->kalkisSaati<<":"<<simdiki->kalkisDakikasi<<"\t\t ";

cout<<simdiki->koltukKapasitesi<<"\t\t\t";

cout<<simdiki->biletFiyati<<"\n";

cout<<"\n\nSatin almak istediginiz bilet sayisini giriniz: ";

cin>>biletSayisi;

yeni->biletSayisi=biletSayisi;

yeni->sonraki = ilk;

ilk = yeni;

simdiki->biletSayisi=yeni->biletSayisi;

while(biletSayisi>10)

{

cout<<"En fazla 10 adet bilet satin alinabilmektedir. Lutfen tekrar deneyiniz: ";

cin>>biletSayisi;

}

if(biletSayisi>simdiki->koltukKapasitesi)

{

cout<<"\nUcusta bu kadar bos koltuk bulunmamaktadir."<<endl;

}

else

{

cout<<"\n"<<" 1. kisinin odeyecegi ucret: "<<simdiki->biletFiyati<<"TL";

if(biletSayisi!=1)

{

for(j=1;j<biletSayisi;j++)

{

float ucret[biletSayisi];

for(i=100;i>=0;i-=10)

{

simdiki->dolulukOrani=(float)((simdiki->koltukKapasitesi)-(simdiki->bosKoltukSayisi))\*100/(simdiki->koltukKapasitesi);

bosKoltukOrani=100-simdiki->dolulukOrani;

a=i-10;

k=k+0.1;

if(i>=bosKoltukOrani && a<bosKoltukOrani)

{

ucret[j]=simdiki->biletFiyati\*k;

cout<<"\n"<<"ilk "<<j+1<<" kisinin odeyecegi ucret: "<<ucret[j]<<"TL";

ucret[j+1]=toplamFiyat-ucret[j];

toplamFiyat+=ucret[j];

}

}

simdiki->bosKoltukSayisi--;

}

}

cout<<"\n\nOdemeyi kabul ediyor musunuz? (kabul ediyorsaniz e yaziniz): ";

cin>>odemeKabul;

cout<<"\n";

bilgi dizi[biletSayisi];

if(odemeKabul=='e')

{

for(int m=0;m<biletSayisi;m++)

{

cout<<m+1<<". bilet sahibinin tc kimlik numarasini giriniz:";

cin>>tcNo;

dizi[m].settcNo(tcNo);

}

simdiki->bosKoltukSayisi=simdiki->koltukKapasitesi-biletSayisi;

simdiki->dolulukOrani=(float)((simdiki->koltukKapasitesi)-(simdiki->bosKoltukSayisi))\*100/(simdiki->koltukKapasitesi);

cout<<"\n\nUcus numarasina ait yeni bilgiler su sekildedir: "<<endl;

cout<<"\n\nUcus No\t\tKalkis Yeri\tVaris Yeri\tZamani\t\tKapasite\tBos Koltuk Sayisi\tDoluluk\t\tBilet Fiyati\n-------\t\t-----------\t----------\t------\t\t--------\t-----------------\t-------\t\t------------\n";

cout<<simdiki->numara<<"\t\t";

cout<<simdiki->kalkisYeri<<"\t\t";

cout<<simdiki->varisYeri<<"\t\t";

cout<<simdiki->kalkisSaati<<":"<<simdiki->kalkisDakikasi<<"\t\t";

cout<<simdiki->koltukKapasitesi<<"\t\t";

cout<<simdiki->bosKoltukSayisi<<"\t\t\t";

cout<<"%"<<simdiki->dolulukOrani<<"\t\t";

cout<<simdiki->biletFiyati<<"\n";

}

else

{

cout<<"\n\nCikis yapiliyor. iyi gunler.";

}

}

}

else

{

cout<<"\nAradiginiz ucusa iliskin kayit bulunmamaktadir.\n";

}

}

void BiletBilgisi(int numara)

{

bilgi\* simdiki = ilk;

float toplam\_kazanc=0,ort\_bilet\_fiyati=0;

while (simdiki != NULL)

{

cout<<"\n\nUcus No\t\tKalkis Yeri\t\tVaris Yeri\t\tZamani\t\tBilet Fiyati\n-------\t\t-----------\t\t----------\t\t------\t\t------------\n";

for(int i=0;i<simdiki->biletSayisi;i++)

{

if(numara==simdiki->numara)

{

cout<<simdiki->numara<<"\t\t";

cout<<simdiki->kalkisYeri<<"\t\t\t";

cout<<simdiki->varisYeri<<"\t\t\t";

cout<<simdiki->kalkisSaati<<":"<<simdiki->kalkisDakikasi<<"\t\t ";

cout<<simdiki->biletFiyati<<"\n";

toplam\_kazanc+=simdiki->biletFiyati;

}

cout<<"\nSatilan Biletler:\nTC Kimlik No\tFiyati:\n-------------\t----------\n";

bilgi dizi[simdiki->biletSayisi];

for(int m=0;m<simdiki->biletSayisi;m++){

cout<<dizi[m].gettcNo();

}

cout<<simdiki->biletFiyati<<"\t\n";

if(simdiki->biletSayisi>0){

ort\_bilet\_fiyati=(float)toplam\_kazanc/simdiki->biletSayisi;

cout<<"--------------------------\n\n";

cout<<"toplam bilet sayisi:"<<simdiki->biletSayisi<<"\t\t\n";

cout<<"toplam kazanc:"<<toplam\_kazanc<<" TL\t\t\n";

cout<<"ortalama bilet fiyati:"<<ort\_bilet\_fiyati<< "TL\t\t\n";

}

else

{

cout<<"Girdiginiz ucus numarasina iliskin satilan bilet yoktur.\n";

}

}

simdiki = simdiki->sonraki;

}

}

void YolcuBilgisi(string tcKimlikNo)

{

bilgi\* simdiki = ilk;

bilgi \*yeni = new bilgi;

bilgi dizi[yeni->biletSayisi];

cout<<"Tc kimlik no giriniz:";

cin>>tcKimlikNo;

while (simdiki != NULL)

{

for(int i=0;i<yeni->biletSayisi;i++)

{

if(Karsilastir(tcKimlikNo)==Karsilastir(dizi[i].gettcNo()))

{

cout<<"\n\nUcus No\t\tKalkis Yeri\t\tVaris Yeri\t\tZamani\t\tBilet Fiyati\n-------\t\t-----------\t\t----------\t\t------\t\t------------\n"<<simdiki->numara<<"\t\t"<<simdiki->kalkisYeri<<"\t\t\t"<simdiki->varisYeri<<"\t\t\t"<<simdiki->kalkisSaati<<":"<<simdiki->kalkisDakikasi<<"\t\t "<<simdiki->biletFiyati<<"\n";

simdiki = simdiki->sonraki;

}

}

}

}

void BiletSil(int numara,string tcKimlikNo,string kalkisYeri,string varisYeri,int kalkisSaati,int kalkisDakikasi)

{

bilet\* simdiki = first;

cout<<"Tc kimlik no giriniz:";

cin>>tcKimlikNo;

while (simdiki != NULL)

{

if(simdiki->TCno==tcKimlikNo)

{

onceki->sonraki = simdiki->sonraki;

delete simdiki;

}

onceki = simdiki;

simdiki = simdiki->next;

}

if(biletSayisi==0)

{

cout<<"\nAradiginiz kriterlere uyan bilet bulunamadi.\n";

}

}

private:

bilet \*first;

};

\*\*/

int main()

{

UcusListe liste;

int numara,secim,biletSayisi,kalkisSaati,kalkisDakikasi,koltukKapasitesi,bosKoltukSayisi,guncelKalkisSaati,guncelKalkisDakikasi;

char devam;

float biletFiyati;

string kalkisYeri,varisYeri,tcNo;

cout<<"\*Havayolu Bilet Satis Sistemine Hos Geldiniz\*\n\n"<<endl;

do

{

cout<<"Yapmak istediginiz islemi seciniz\n\n"<<endl;

cout<<" 1 - Yeni bir ucus ekle\n 2 - Ucusun kalkis zamanini guncelle\n 3 - Ucusa iliskin bilet satin al\n 4 - Ucusu iptal et\n 5 - Satilan bileti iptal et\n 6 - Bir yerden kalkan ucuslarin bilgilerini listele\n 7 - Bir yerden bir yere olan ucuslarin bilgilerini listele\n 8 - Bir ucusun bilgilerini ve o ucusa iliskin satilan biletleri listele\n 9 - Koltuk doluluk orani %50'nin altinda olan ucuslari listele\n 10 - Bir yolcunun biletlerini listele\n\n:";

cin>>secim;

while(secim<1||secim>10)

{

cout<<"\nSeciminiz 1 ile 10 sayilari arasinda bir tam sayi olmalidir, lutfen tekrar deneyiniz:";

cin>>secim;

}

if(secim<=5||secim==8||secim==10)

{

cout<<"\nUcus numarasi giriniz:";

cin>>numara;

while(numara<1000||numara>9999)

{

cout<<"Ucus numarasi 1000-9999 arasi bir tam sayi olmalidir, lutfen tekrar deneyiniz:";

cin>>numara;

cout<<"\n";

}

}

switch(secim)

{

case 1:

liste.UcusEkle(numara,kalkisYeri,varisYeri,kalkisSaati,kalkisDakikasi,koltukKapasitesi,biletFiyati);

break;

case 2:

liste.ZamanGuncelle(numara);

break;

case 3:

cout<<"\nBu fonksiyon duzenlenmektedir.\n";

break;

case 4:

liste.UcusSil(numara);

break;

case 5:

cout<<"\nBu fonksiyon duzenlenmektedir.\n";

break;

case 6:

liste.KalkisYerineGore(kalkisYeri);

break;

case 7:

liste.KalkisVeVarisYerineGore(kalkisYeri,varisYeri);

break;

case 8:

liste.UcusBilgisi(numara);

break;

case 9:

liste.YuzdeElliAlti();

break;

case 10:

cout<<"\nBu fonksiyon duzenlenmektedir.\n";

break;

}

cout<<"\n\n\nPrograma devam etmek icin 'e'yi, cikis yapmak icin herhangi bir tusu tuslayiniz.";

cin>>devam;

cout << endl;

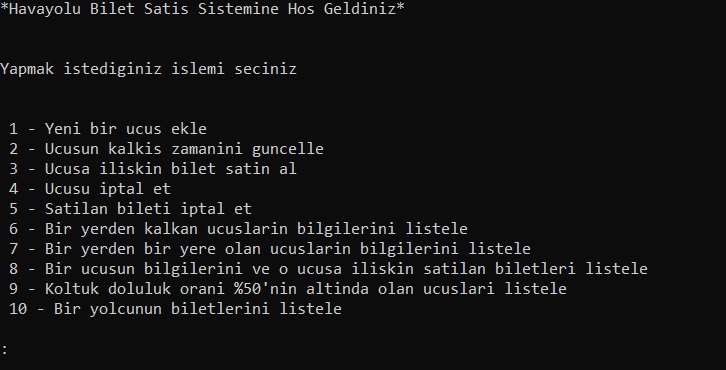
}while((devam)=='e'||(devam)=='E');

cout<<"\n\n\n\t\tTesekkur ederiz!";

return 0;

}

**KULLANICI KATALOĞU**

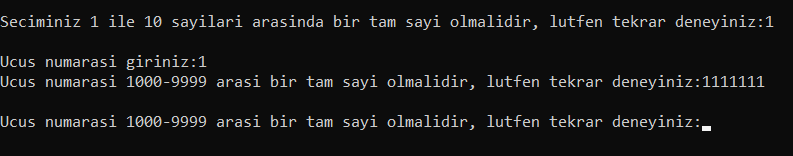


Program çalıştırıldığında karşınıza çıkacak olan bu ekranda yapmak istediğiniz işlemi ilgili sayıyı tuşlayarak seçmeniz gerekmektedir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Seçiminiz 1 ile 10 sayıları arasında değilse program istenilen kıstasa uygun sayı girilene kadar hata verecek, tekrar seçim yapmanızı isteyecektir.



Seçiminizden sonra uçuş numarası girmeniz isteniyorsa 1000 ve 9999 arası bir sayı girmeniz gerekmektedir. Aksi takdirde program uygun sayı girilene kadar hata verecek, tekrar uçuş numarası isteyecektir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yukarıdaki seçimi yaptığınızda çıkan soruları nasıl yanıtlayabileceğinize dair bir örnek gösterilmiştir. İstenilen soruları benzer şekilde yanıtlayarak yeni bir uçuş ekleyebilirsiniz.

metin, monitör, siyah, ekran görüntüsü içeren bir resim

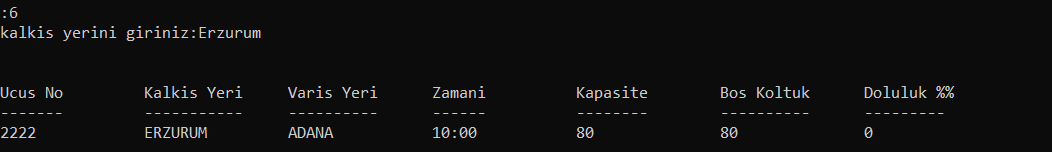
Açıklama otomatik olarak oluşturuldu

Yukarıdaki seçimi yaptığınızda çıkan soruları nasıl yanıtlayabileceğinize dair bir örnek gösterilmiştir. İstenilen soruları benzer şekilde yanıtlayarak uçuş zamanınızı güncelleyebilirsiniz.

Yukarıdaki seçimi yaptığınızda çıkan soruları nasıl yanıtlayabileceğinize dair bir örnek gösterilmiştir. İstenilen soruları benzer şekilde yanıtlayarak bilet alım işleminizi gerçekleştirebilirsiniz.



Yukarıdaki seçimi yaptığınızda çıkan ekran gösterilmiştir. Silmek istediğiniz uçuşu numarasını girerek silebilirsiniz.

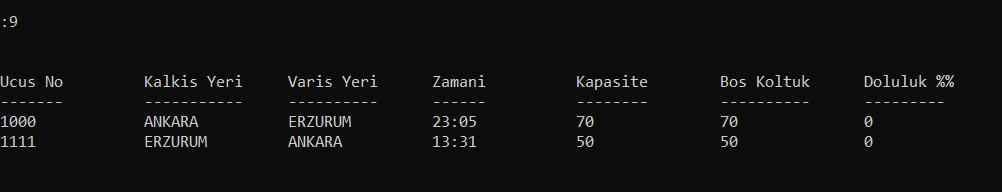


Yukarıdaki seçimi yaptığınızda çıkan soruları nasıl yanıtlayabileceğinize dair bir örnek gösterilmiştir. İstenilen soruları benzer şekilde yanıtlayarak istediğiniz yerden kalkan uçuşları görebilirsiniz.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yukarıdaki seçimi yaptığınızda numaralandırılmış seçimi yaptığınızda çıkan soruları nasıl yanıtlayabileceğinize dair bir örnek gösterilmiştir. İstenilen soruları benzer şekilde yanıtlayarak istediğiniz rotada ilerleyen uçuşları görebilirsiniz.



Yukarıda numaralandırılmış seçimi yaptığınızda karşınıza çıkabilecek örnek bir ekran gösterilmiştir. Bu seçiminizle doluluk oranı yüzde elli altında olan uçuşları görüntüleyebilirsiniz.