**T.C.  
ATATÜRK ÜNİVERSİTESİ**

**NESNEYE YÖNELİK PROGRAMLAMA**

**PROJE-3**

**HAVAYOLU BİLET SATIŞ SİSTEMİ**

**Khadije FAKSH  
220710052**

28.05.2023

**ERZURUM 2023**

**İçindekiler**

[**1.ANALİZ 3**](#_Toc131268754)

[**2.TASARIM 4**](#_Toc131268755)

[**3.PROGRAMCI KATALOĞU 9**](#_Toc131268756)

[**4.KULLANICI KATALOĞU 21**](#_Toc131268757)

**1. SİSTEMİN ANALİZ KISMI**

Bu projede nesneye yönelik programlama (OOP) yöntemleri kullanılarak bir havayolu şirketinin günlük uçuşlarıyla ilgili işlem yapmasını sağlayan uygulama istenmektedir. Bu amaçla dört sınıf oluşturulacaktır. Bu sınıflar Uçuş, Uçuş numarasına göre uçuş listesi, kalkış yerine göre uçuş listesi ve string’dir. Bu işlemler uçuş eklenmesi, güncellenmesi, iptal edilmesi ve görüntülenmesi şeklinde özetlenebilir.- Birazdan bu işlemler ayrıntılı bir şekilde anlatılacaktır.- Bahsedilen işlemlerin gerçekleşmesi için hem uçuş numaraları hem de kalkış yerleri aracılığıyla uçuş numarası, kalkış yeri, varış yeri, kalkış zamanı, normal koltuk sayısı ve lüks koltuk sayısına erişilmesi istenmektedir. Uçuş bilgilerine uçuş numarası aracılığıyla erişmek için uçuş numarasına göre küçükten büyüğe doğru sıralı olan bir tek bağlı liste kullanılacaktır. Uçuş bilgilerine kalkış yeri aracılığıyla erişmek içinse kalkış yerine göre sıralı-kalkış yeri aynı ise de kalkış zamanına göre sıralı- olan bir çift bağlı liste kullanılacaktır. Kodun içereceği işlemler ise şu şekildedir:

1) Lüks Koltuk İçermeyen Yeni Bir Uçuşun Eklenmesi:  
Uçuş numarası kullanıcıdan alınacaktır, böyle bir uçuş numarası yok ise uçuş numarası, kalkış yeri, varış yeri, kalkış zamanı ve normal koltuk sayısı alınmalıdır. Var ise ekleme yapılmamalıdır.  
2) Lüks Koltuk İçeren Yeni Bir Uçuşun Eklenmesi:   
Uçuş numarası kullanıcıdan alınacaktır, böyle bir uçuş numarası yok ise uçuş numarası, kalkış yeri, varış yeri, kalkış zamanı, normal koltuk sayısı ve lüks koltuk numarası alınmalıdır. Var ise ekleme yapılmamalıdır.  
3) Bir Uçuşun Lüks Koltuk Sayısının Güncellenmesi:  
Uçuş numarası kullanıcıdan alınacaktır, böyle bir uçuş numarası varsa lüks koltuk sayısı güncellenmelidir. Aksi takdirde böyle bir uçuşun olmadığı belirtilmelidir.  
4) Bir Uçuşun İptal Edilmesi:  
Uçuş numarası kullanıcıdan alınacaktır, böyle bir uçuş numarası var ise bu uçuşun uçuş numarası, kalkış yeri, varış yeri, kalkış zamanı, normal koltuk sayısı ve lüks koltuk sayısı silinmelidir. Aksi takdirde böyle bir uçuşun olmadığı belirtilmelidir.  
5) Bir Uçuşun Bilgilerinin Görüntülenmesi:  
Uçuş numarası kullanıcıdan alınacaktır, böyle bir uçuş numarası var ise bu uçuşun uçuş numarası, kalkış yeri, varış yeri, kalkış zamanı, normal koltuk sayısı ve lüks koltuk sayısı listelenmelidir. Böyle bir uçuş yoksa bu bilgi kullanıcıya belirtilmelidir.  
6) Lüks Koltuk İçeren Uçuşların Listelenmesi:  
Lüks koltuk içeren tüm uçuşların bilgileri, uçuş numaralarına göre artarak sıralanmalıdır.  
7) Bir Yere Varan Uçuşların Listelenmesi:  
Varış yeri kullanıcıdan alınacaktır, böyle bir kalkış yeri var ise bu varış yerine olan uçuşların uçuş numarası, kalkış yeri, varış yeri, kalkış zamanı, normal koltuk sayısı ve lüks koltuk sayısı kalkış yerlerini alfabetik şekilde sıralayarak vermelidir.  
8) Bir Yerden Kalkan Uçuşların Listelenmesi:  
Kalkış yeri kullanıcıdan alınacaktır, böyle bir kalkış yeri var ise bu varış yerine olan uçuşların uçuş numarası, kalkış yeri, varış yeri, kalkış zamanı, normal koltuk sayısı ve lüks koltuk sayısı kalkış zamanlarına göre artarak sıralanmalıdır.  
9)Bir Yerden Kalkan ve Bir Yere Varan Uçuşların Listelenmesi:  
Varış yeri ve kalkış yeri kullanıcıdan alınacaktır, bu kalkış yerinden kalkan ve bu varış yerine varan uçuşların bilgileri kalkış zamanları azalan sırada verilmelidir.

**2.TASARIM**

**Ucus Sınıfı:**

#pragma once

#include <iostream>

using namespace std;

class Ucus

{

private:

int ucus\_numarasi;

string kalkis\_yeri;

string varis\_yeri;

string kalkis\_zamani;

int normal\_koltuk;

int luks\_koltuk;

Ucus\* next; :Bir sonraki düğümü işaretlemek için tanımlanmıştır.

public:

Ucus();//constructor

void ucusEkle();

void luksKoltukBilgi();

int getUcus\_numarasi();

string getKalkis\_yeri();

string getVaris\_yeri();

string getKalkis\_zamani();

int getNormal\_koltuk();

int getLuks\_koltuk();

Ucus\* getNext();

void ucusno();  
 void setLuks(int );

void setNext(Ucus\*);

~Ucus(); //destructor  
};

* **Ucus() Fonksiyonu:**

Yukarıda verilen değişkenlerin ilk atamasını yapar. Constructor’dur.

* **ucusno() Fonksiyonu:**cout ile Uç Numarası mesajını yazdırır ve uç numarasını alır uçuş numarası 0 olduğu sürece uçuş numarasının 0 olmaması gerektiğini yazdırır ve tekrar alır.
* **ucusEle() Fonksiyonu:**

cout’la cout’ tan sonra hangi değişkeni alacaksa onu söyler ve ondan sonra getline ile alır. Arada tamponu temizlemek için cin.ignore() fonksiyonunu kullanır. Eklenilen varış yeri kalkış yerine eşit olduğu sürece bu hatayı belirten bir mesaj yazdırır. Kalkış zamanını stoi() fonksiyonuyla string olarak alır ve geçerli bir saat olup olmadığnı kontrol eder.

* **luksKoltukBilgi() Fonksiyonu:** Kullanıcıdan lüks koltuk sayısını alır. İlgili mesajı ekrana yazdırır ve kullanıcının girişini cin ile okur. Girilen değeri luks\_koltuk değişkenine atar. Eğer lüks koltuk sayısı normal koltuk sayısından büyük ise, kullanıcıya normal koltuk sayısına eşit veya daha küçük bir değer girmesi gerektiği mesajını verir ve tekrar bir değer alınır.
* **getUcus\_numarasi() Fonksiyonu:** uçuş numarası değişkenini geri döndürür.
* **getKalkis\_yeri() Fonksiyonu:** kalkış yeri değişkenini geri döndürür.
* **getKalkis\_zamani() Fonksiyonu:** kalkış zamanı değişkenini geri döndürür.
* **getNormal\_koltuk() Fonksiyonu:** normal koltuk değişkenini geri döndürür.
* **getLuks\_koltuk() Fonksiyonu:** luks koltuk değişkenini geri döndürür.
* **getNext() Fonksiyonu:** next işaretçisini geri döndürür.
* **setLuks(int koltuk) Fonksiyonu:** lüks koltuk değişkenine koltuğu atar
* **setNext(Ucus\* next) Fonksiyonu:** next işaretçisine next değerini atar.

**Ucus\_Numarasına\_Gore\_Ucus Sınıfı:**

#pragma once

#include "Ucus.h"

class Ucus\_Numarasina\_Gore\_Ucus\_Listesi

{

private:

Ucus\* head; //uçuş tipinde işaretçi bir head oluşturulmuştur.

public:

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

void Ekle(int);

void luksUcus();

void luksGuncelleme();

void goruntuleme();

void ucusiptal();

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

};

* **Ucus\_Numarasina\_Gore\_Ucus\_Listesi() Fonksiyonu:**Ucus\_Numarasina\_Gore\_Ucus\_Listesi sınıfının kurucu fonksiyonudur. Başlangıçta head işaretçisi nullptr olarak atanır.
* **Ekle(int secim1) Fonksiyonu:**Uçuş numarasına göre uçuş listesine yeni bir uçuş ekler. Kullanıcıdan uçuş bilgilerini alır ve yeni bir Ucus nesnesi oluşturur. Ardından, eklenmek istenen uçuş numarasının zaten listede olup olmadığını kontrol eder. Eğer aynı uçuş numarası zaten mevcutsa, kullanıcıdan uygun bir uçuş numarası girmesini ister. secim1 parametresine göre lüks koltuk içeren bir uçuş eklenmek isteniyorsa, luksKoltukBilgi() fonksiyonu çağrılır. Daha sonra, uçuş listesinde yeni uçuşun doğru konuma eklenmesi sağlanır.
* **luksUcus() Fonksiyonu:**Bu fonksiyon, lüks koltuk içeren uçuşları listeler. İlk olarak, head işaretçisinin nullptr olup olmadığını kontrol eder. Eğer liste boşsa, kullanıcıya hiçbir uçuşun eklenmediği mesajı verilir ve fonksiyon sonlandırılır. Eğer liste boş değilse, current2 işaretçisi head işaretçisine atanır. Ardından, setw() ve left manipülatörlerini kullanarak formatlı bir çıktı oluşturulur ve başlık bilgileri yazdırılır. Daha sonra, current2 işaretçisi nullptr olmadığı sürece döngü devam eder. Her döngü adımında, current2 işaretçisinin luks\_koltuk değeri kontrol edilir. Eğer lüks koltuk sayısı sıfır değilse, ilgili uçuşun bilgileri formatlı bir şekilde yazdırılır. Son olarak, current2 işaretçisi bir sonraki uçuşa geçirilir.
* **luksGuncelleme() Fonksiyonu:**

Uçuş numarasına göre lüks koltuk sayısını günceller. Kullanıcıdan güncellenmek istenen uçuş numarasını ve yeni lüks koltuk sayısını alır. Ardından, uçuş listesinde ilgili uçuş numarasını arar ve bulduğu durumda lüks koltuk sayısını günceller. Eğer uçuş numarası bulunamazsa, kullanıcıya uygun bir uçuş numarası girmesi istenir.

* **goruntuleme() Fonksiyonu:**

Kullanıcıdan bir uçuş numarası (ucusNo8) alarak, bu numaraya sahip olan uçuşu liste içerisinde arar ve ekrana bilgilerini yazdırır.  
İlk olarak, head işaretçisinin nullptr olup olmadığı kontrol edilir. Eğer liste boşsa, kullanıcıya henüz uçuş eklenmediği mesajı verilir ve fonksiyon sonlandırılır.  
Eğer liste boş değilse, kullanıcıdan görüntülemek istediği uçuş numarasını alır (ucusNo8).  
Daha sonra, bir döngü kullanarak liste içerisindeki uçuşları kontrol eder. Döngü, current10 işaretçisi nullptr olmadığı ve current10->getUcus\_numarasi() değeri ucusNo8'ye eşit olmadığı sürece devam eder. Her döngü adımında, current10 işaretçisi bir sonraki uçuşa ilerletilir.  
Döngü sonlandığında, eğer current10 işaretçisi nullptr ise, aranan uçuş bulunamadığı anlamına gelir. Bu durumda, kullanıcıya "Böyle bir uçuş bulunamadı!" mesajı verilir ve fonksiyon sonlandırılır. Eğer current10 işaretçisi nullptr değilse, aranan uçuş bulunmuştur. Bu durumda, formatlı bir çıktı   
oluşturulur ve uçuşun bilgileri (current10 işaretçisi üzerinden) yazdırılır.  
Başlık bilgileri (setw() ve left manipülatörleri kullanılarak) ve uçuş bilgileri ekrana yazdırılır.  
Sonuç olarak, bu fonksiyon kullanıcının belirttiği uçuş numarasına sahip olan uçuşun bilgilerini ekrana yazdırır veya uçuşun bulunamadığını bildirir.

* **ucusiptal() Fonksiyonu:**Bu fonksiyon, kullanıcıdan bir uçuş numarası (UcusNo2) alarak, bu numaraya sahip olan uçuşu listeden siler.İlk olarak, head işaretçisinin nullptr olup olmadığı kontrol edilir. Eğer liste boşsa, kullanıcıya henüz uçuş eklenmediği mesajı verilir ve fonksiyon sonlandırılır.  
  Eğer liste boş değilse, kullanıcıdan iptal etmek istediği uçuşun numarasını alır (UcusNo2).  
  Daha sonra, kontrol edilmek üzere bir current7 işaretçisi head işaretçisine atanır.  
  Eğer current7->getUcus\_numarasi() değeri UcusNo2'ye eşitse ve current7 işaretçisi head işaretçisine eşitse, yani listenin başındaki uçuş silinecekse, head işaretçisi current7->getNext() ile güncellenir ve ardından current7 işaretçisi silinir (delete current7).Eğer yukarıdaki koşul sağlanmazsa, bir döngü kullanarak uçuşları kontrol eder. Döngü, current7->getNext() değeri nullptr olmadığı ve current7->getNext()->getUcus\_numarasi() değeri UcusNo2'ye eşit olmadığı sürece devam eder. Her döngü adımında, current7 işaretçisi bir sonraki uçuşa ilerletilir.Döngü sonlandığında, eğer current7->getNext() işaretçisi nullptr ise, aranan uçuş bulunamamış demektir. Bu durumda, kullanıcıya "Böyle bir uçuş bulunamadı!" mesajı verilir ve fonksiyon sonlandırılır.Eğer current7->getNext() işaretçisi nullptr değilse, aranan uçuş bulunmuştur. Bu durumda, silme işlemi gerçekleştirilir. Önce current8 işaretçisi, silinecek uçuşun bir sonraki uçuşunu (current7->getNext()->getNext()) gösterir. Ardından, current7->getNext() işaretçisi silinir (delete current7->getNext()) ve current7 işaretçisinin bir sonraki uçuşu current8 ile güncellenir (current7->setNext(current8)).  
  onuç olarak, bu fonksiyon kullanıcının belirttiği uçuş numarasına sahip olan uçuşu listeden siler veya uçuşun bulunamadığını bildirir.

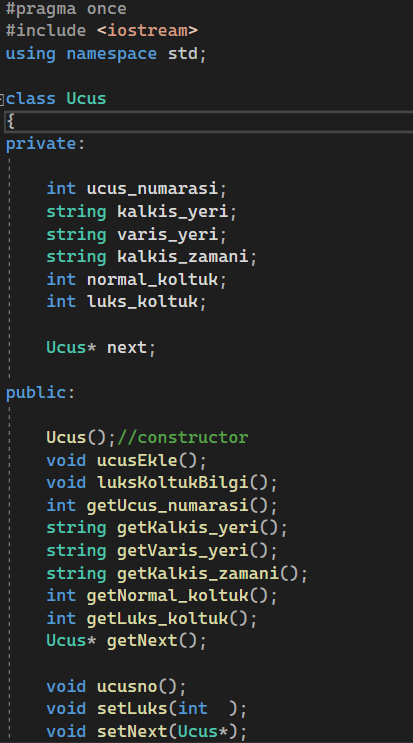
* **~Ucus\_Numarasina\_Gore\_Ucus\_Listesi() Fonksiyonu:**Bu fonksiyon, Ucus\_Numarasina\_Gore\_Ucus\_Listesi sınıfının yıkıcı fonksiyonudur. head işaretçisi üzerinden listenin başından başlayarak tüm uçuş düğümlerini serbest bırakır (delete) ve bellekten siler. Her adımda, current işaretçisi bir sonraki uçuş düğümüne geçirilir.

**3.PROGRAMCI KATALOĞU**

1)Analize Harcanan Süre:6 saat

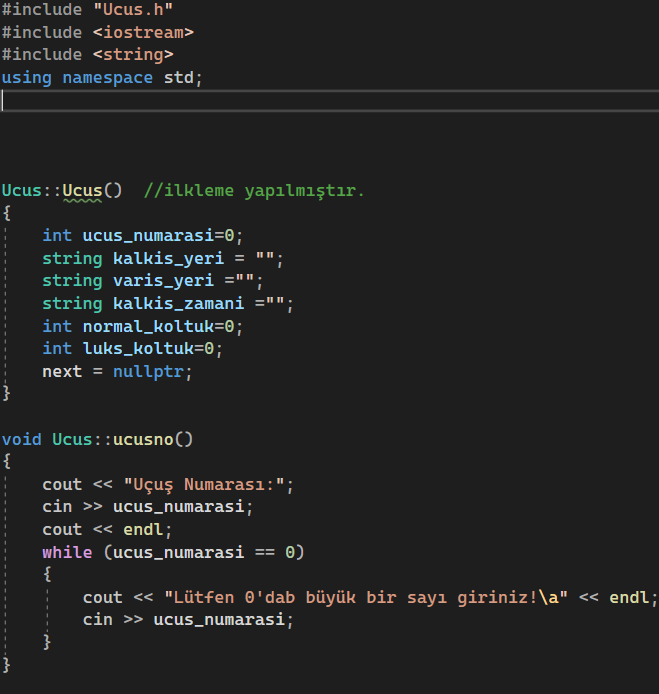
2)Tasarım ve Gerçekleştirime Harcanan Süre:40 saat

3)Test ve Raporlama İçin Harcana Süre: 5 saat

4)Kaynak Kodun Çıktısı:

yazı tipi, grafik, metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu



metin, ekran görüntüsü, yazılım içeren bir resim

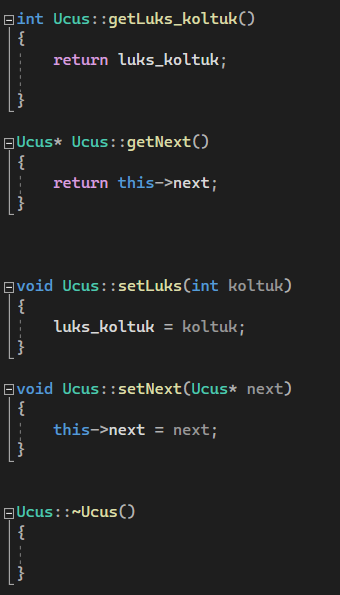
Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

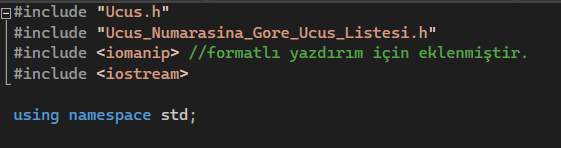
metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu



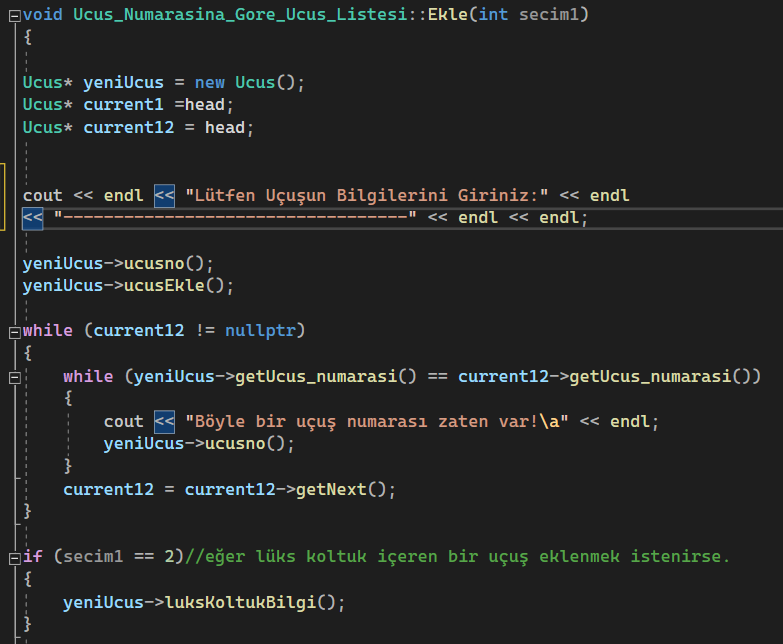
metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu



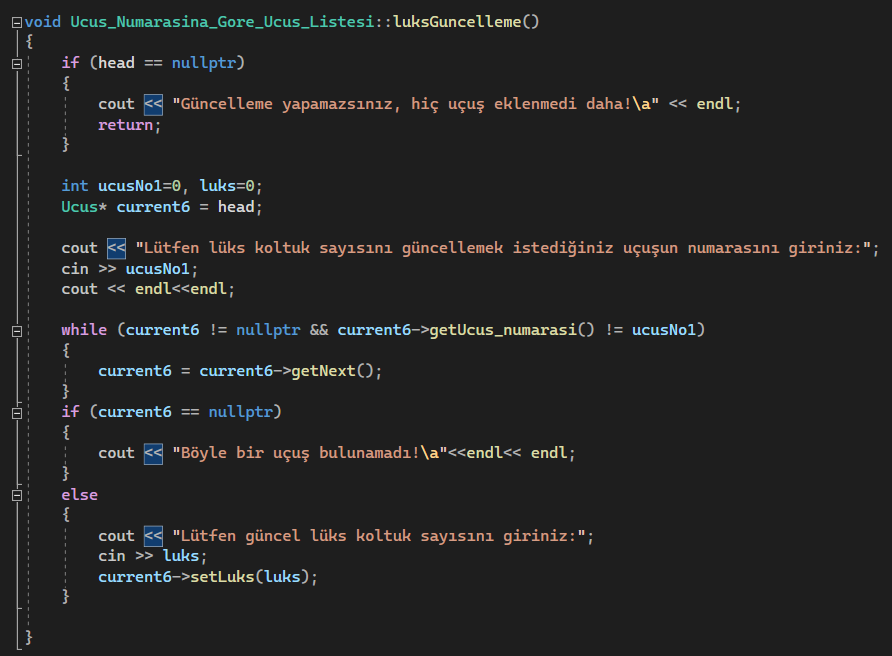
metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu



metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu



metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

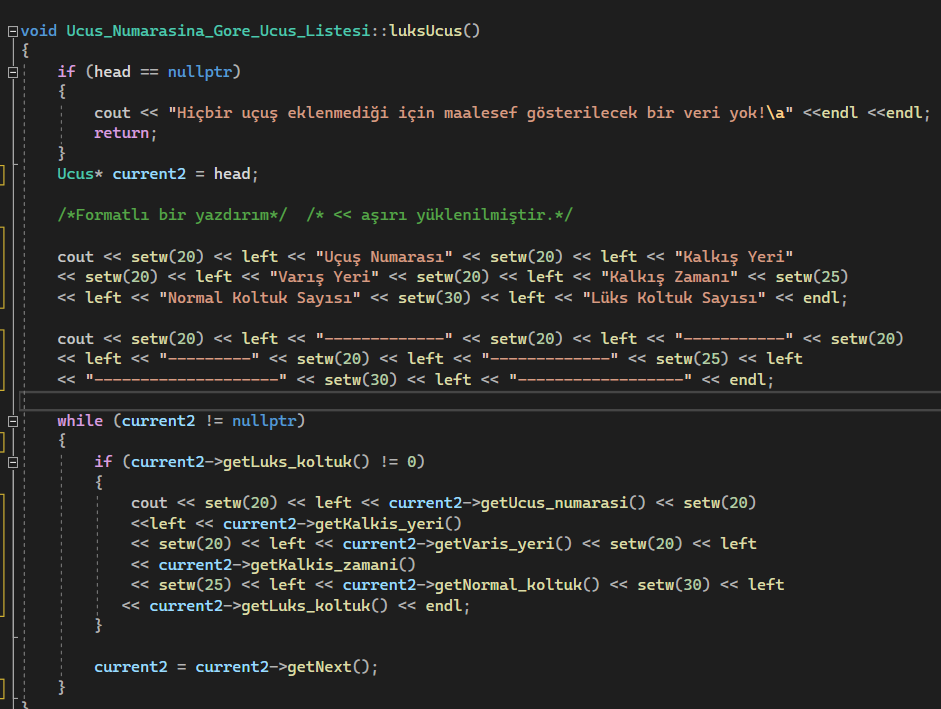
Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

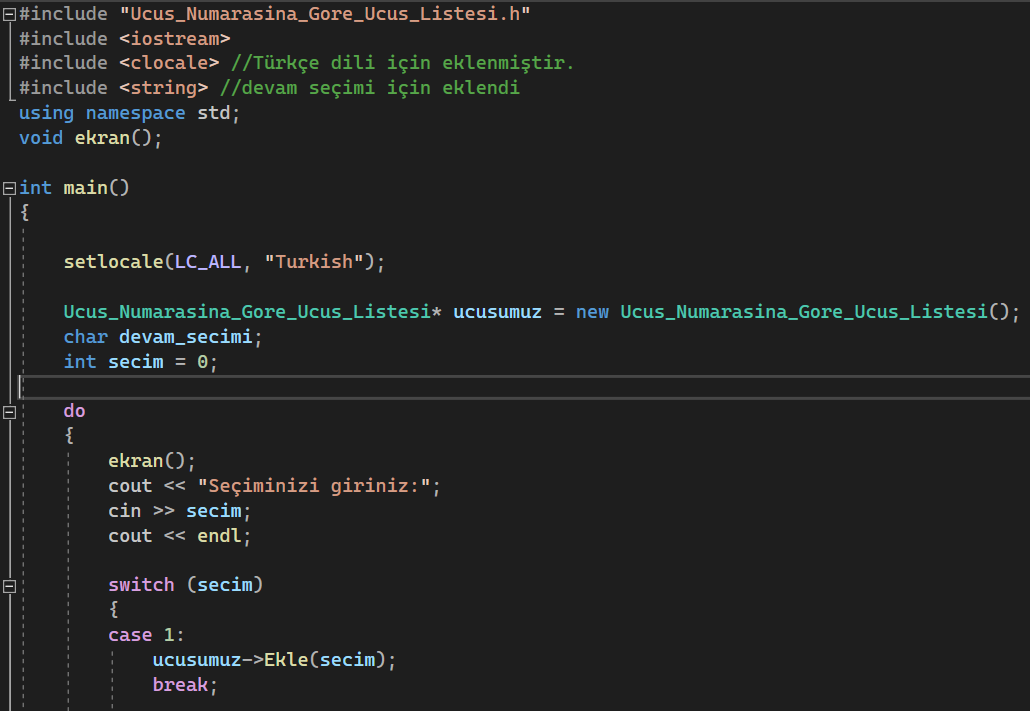
metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu



metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu



metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

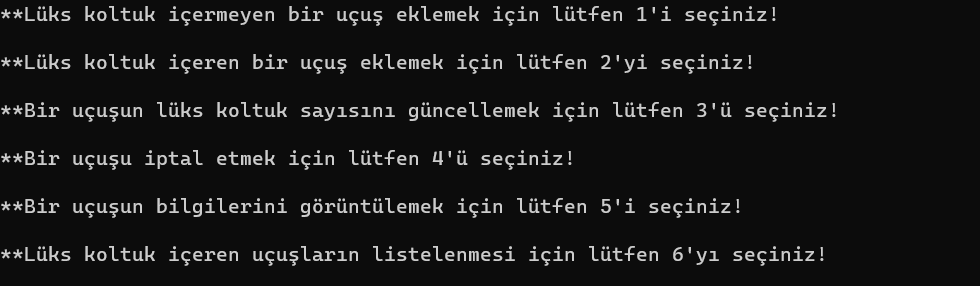
Açıklama otomatik olarak oluşturuldu

**metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu**

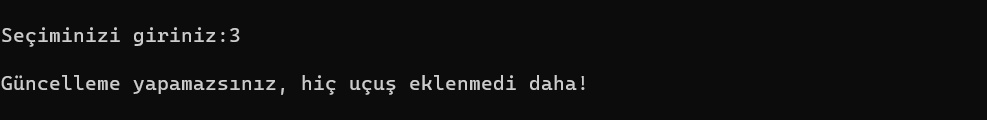
**4.KULLANICI KATALOĞU**

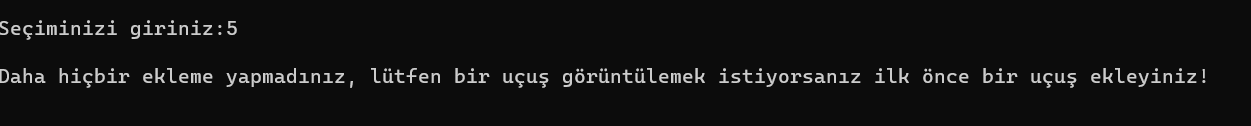
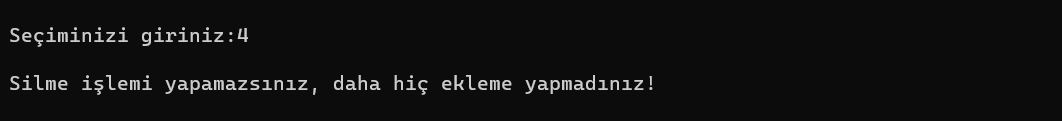
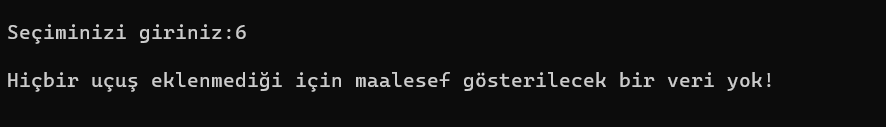
1) Bilgilendirme Mesajı:



Programı açar açmaz önünüze gelecektir ve işlemler hakkında gördüğünüz gibi bilgi verecektir.

2) Kullanım Uyarıları:

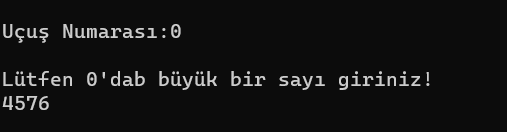




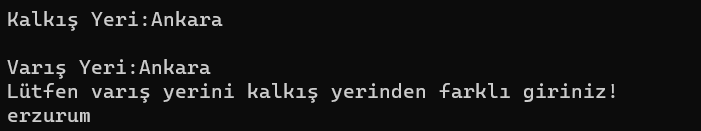
Programa girer girmez bir uçuş eklemeden (1. ve 2.işlem) 3,4,5 ve 6’dan herhangi birini seçerseniz yukarıdaki gibi hata mesajı alırsınız.

3) Dikkat Edilmesi Gerekenler:

1.Uçuş numarasının sıfır olmaması

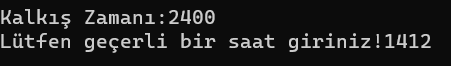


Buna dikkat etmezseniz yukarıdaki gibi bir hata mesajıyla karşılaşırsınız.

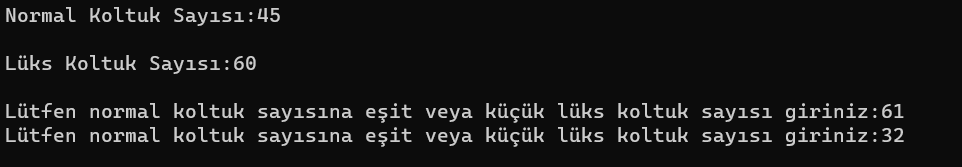
2.Varış yerinin kalkış yeriyle aynı olmaması

Bir uçuş takdir edersiniz ki aynı yerden aynı yere gidemez 😊  
Onun için varış yeriyle kalkış yerini eşit girerseniz bu şekilde hata alacaksınız.  
Farklı bir varış yeri girmediğiniz sürece programımız, farklı bir değer istemeye  
devam edecektir. Haberiniz olsun 😊

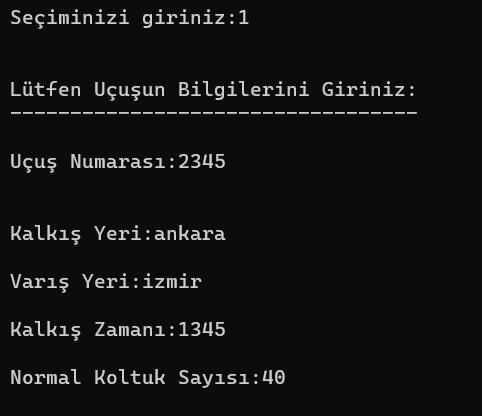
3.Kalkış Zamanının Doğru Girilmesi



Yukarıdaki hatayı almamak için lütfen 24:00’dan küçük değerler giriniz.

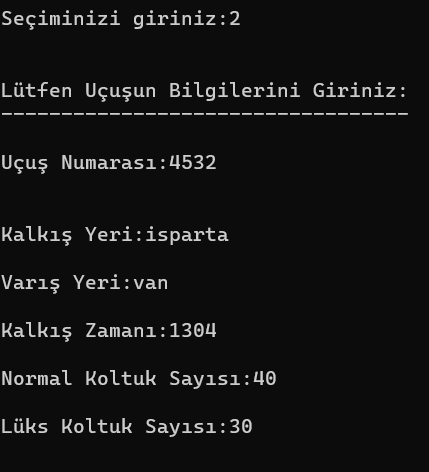
4.Normal Koltuk-Lüks Koltuk

Lüks koltuk sayısı, normal koltuk sayısından küçük veya normal koltuk sayısına eşit olmalıdır. Büyük girerseniz uygun bir değer yazana kadar sizden doğru bir değer istemeye devam edecektir.

4)1.Seçim:

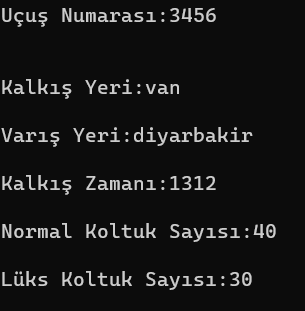
Gördüğünüz gibi 1.seçimde lüks koltuk içermeyen bir uçuş eklemiş olursunuz.

5)2.Seçim:



2.Seçimde ise lüks koltuk sayısı içeren bir uçuş eklemiş olursunuz.

6)3.Seçim:

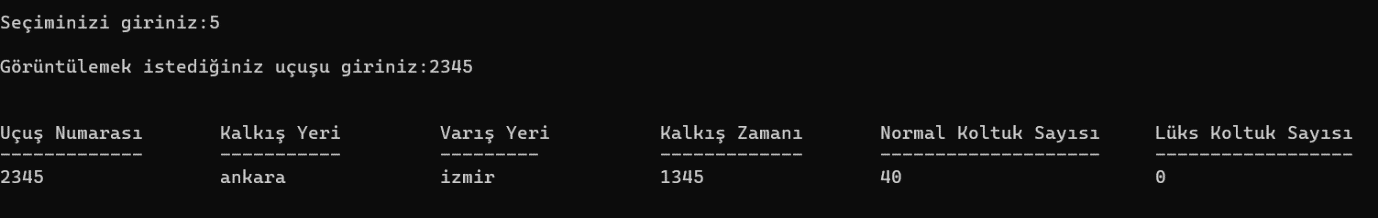


Diyelim ki sizin yandaki örnekteki gibi lüks   
 koltukları olan bir uçuşunuz var. Siz de lüks  
 koltuk sayısını güncellemek istiyorsunuz   
 o zaman 3.seçim imdadınıza yetişiyor.

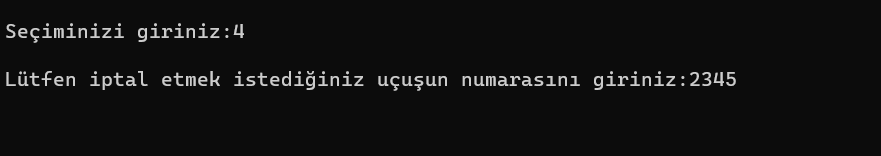
metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

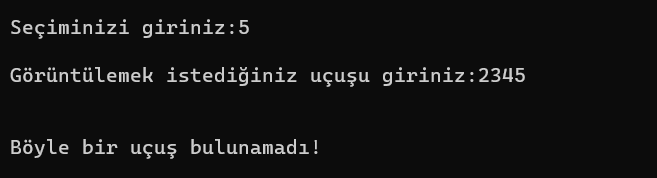
Açıklama otomatik olarak oluşturuldu

5.Seçimi seçip uçuş numarasını girerseniz lüks koltuk sayısının değiştiğini göreceksiniz

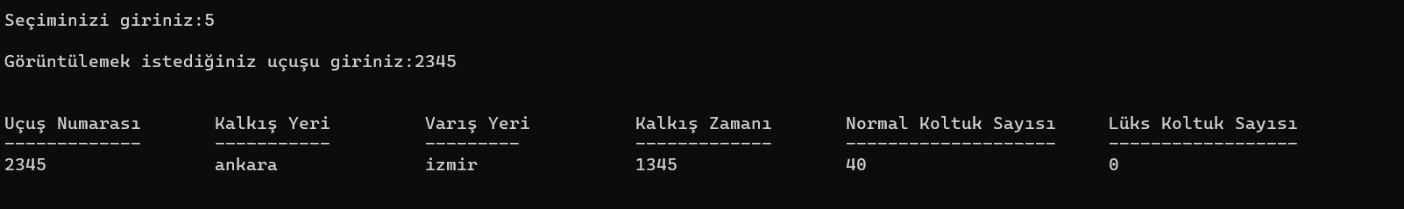
7) 4.Seçim:

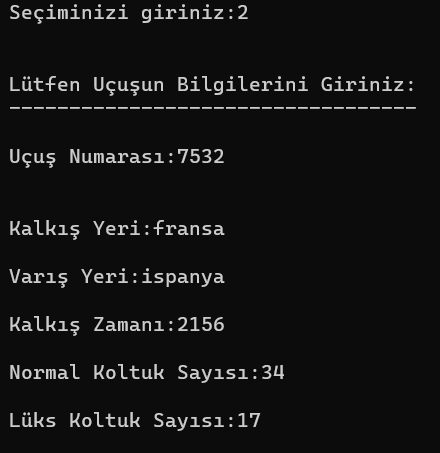
Diyelim ki bir uçuşu silmek istiyorsunuz. Bu işlemi silmek için 4.seçimi kullanırsınız. Yukarıda var olan uçuşun 4.seçimi seçerek nasıl silineceğini görelim:



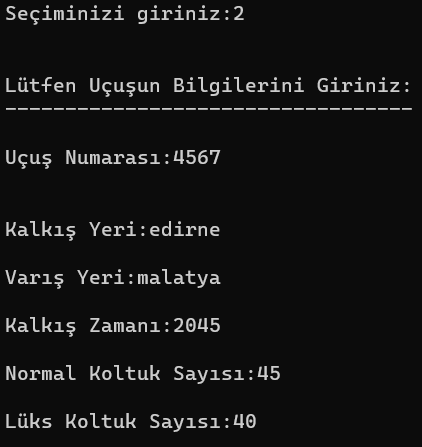
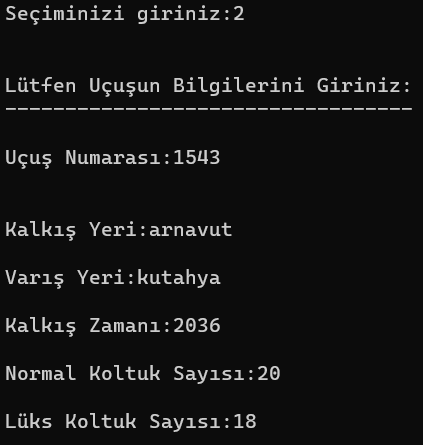
8)5.Seçim:

5.seçim girdiğiniz uçuş numarasının uçuş bilgilerini gösterir. 4.seçimde 2345 numaralı uçuş silindiği içim bu uçuşu bulamayacaktır. Var olan bir uçuşu aratırsanız aşağıdaki gibi bir sonuç verecektir:

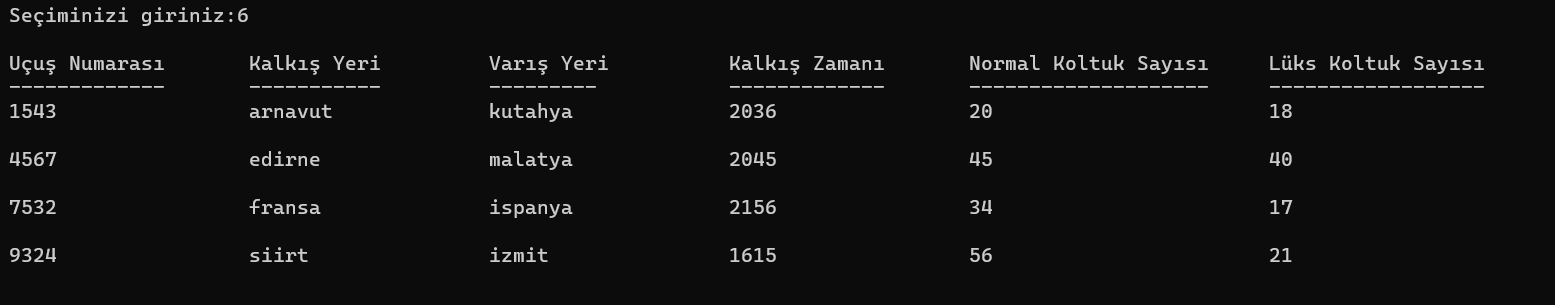


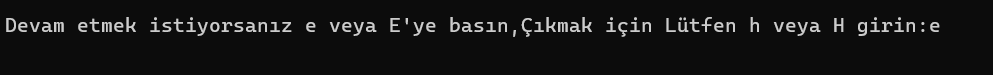
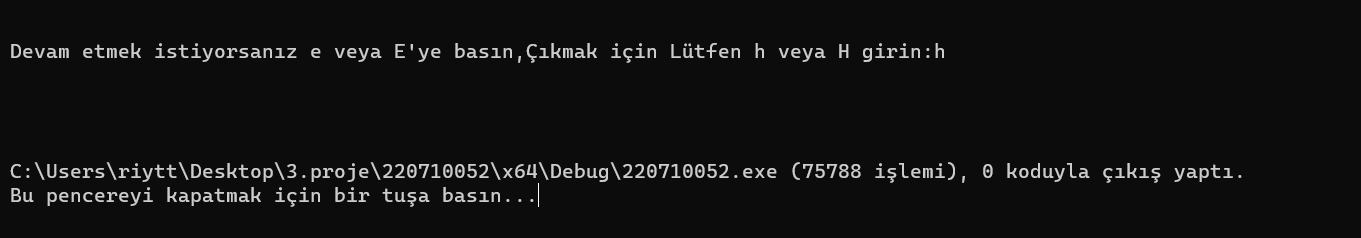
9)6.Seçim:

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sırayla bu uçuşları eklemiş olun. 6.Seçim bu uçuşları (lüks koltuğa sahip uçuşları) uçuş numaralarına göre küçükten büyüğe doğru sıralar:



Başka Seçim Yapmak İsterseniz ‘ e’ veya ‘ E ‘

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu Başka Bir Şey Girerseniz Size Hata Mesajı Verecektir

Çıkmak İsterseniz ‘ h’ veya ‘ H’ basınız