



Nesneye Yönelik Programlama

Havayolu Ucus Takip Sistemi

Mahmud Elmuhammed-220710078

Teslim Tarihi: 27-05-2023



Programın Amacı

->Programımızın çalışmasıyla kullanıcı aşağıdaki işlemleri yapabilecektir:

- 1)Sisteme lüks koltuk içermeyen yeni uçuş ekleme
- 2)Sisteme lüks koltuk içeren yeni uçuş ekleme
- 3) Bir uçuşun lüks koltuk sayısını güncellenmesi
- 4) Bir uçuşun iptal edilmesi
- 5) Bir uçuşun bilgilerini görüntülenmesi
- 6)Lüks koltuk içeren uçuşların listelenmesi
- 7) Bir yere olan uçuşların listelenmesi
- 8) Bir yerden kalkan uçuşların listelenmesi
- 9) Bir yerden bir yere olan uçuşların listelenmesi

Kullanıcının kataloğu

```
1. Luks koltuk icermeyen yeni bir ucus ekle
2. Luks koltuk iceren Yeni bir ucus ekle
3. Bir ucusun luks koltuk sayisini guncellenmesi
4. Bir ucusun iptal edilmesi
5. Bir ucusun bilgilerini görüntülenmesi
6. Luks koltuk iceren ucularin listelenmesi
7. Bir yere olan ucusların listelenmesi
8. Bir yerden kalkan ucusların listelenmesi
9. Bir yerden bir yere olan ucusların listelenmesi

yapacağınız işlem numarası lütfen girin:
```

Sistemin arayüzü ve yapmak istediğiniz işlemin numarasının girmeniz istenir.

1)Sisteme lüks koltuk içermeyen yeni uçuş ekleme

->Ucus numarasını girin:

ucus numarası önceden kayıt olmaması lazım.

->Kalkis yeri girin:

->Varis yeri girin:

•->Kalkis zamanı girin: 12:12

kalkış zamanı bu şekilde (12:12) girmesi lazım.

->Normal koltuk sayisini girin:

•baska bir işlem yapmak ister misiniz? (1/0):
girin.

devam etmek için 1 girin bitirmek için 0

2)Sisteme lüks koltuk içeren yeni uçuş ekleme

- Bu işlemin 1. işlemin aynısı tek farkı bu işlemde lüks koltuk sayısı istenir.

`->Lüks koltuk sayisini girin:`

3) Bir uçuşun lüks koltuk sayısını güncellenmesi

- `->Ucus numarası girin:` güncellemek istediğiniz uçuş numarası.

`->Yeni lüks sayisini girin:`

4) Bir uçuşun iptal edilmesi

- `->Ucus numarası girin:` //bu işlem tamamen çalışmıyor.

5) Bir uçuşun bilgilerini görüntülenmesi

- `->Ucus numarası girin:` görüntülemek istediğiniz uçuş numarası.

- Örnek:

```
->Ucus numarası girin: 122
->Ucus Bilgileri:
    ->Ucus Numarası: 122
    ->Kalkış Yeri: a
    ->Varış Yeri: c
    ->Kalkış Zamanı: 12:45
    ->Normal Koltuk Sayısı: 5
    ->Lüks Koltuk Sayısı: 9
```

Bu şekilde uçuş bilgileri yazdıracak.

6) Lüks koltuk içeren uçuşların listelenmesi

- Bu işlemde direk lüks koltuk içeren uçuşlar yazdıracak.

```
Luks koltuk iceren ucularin listelenmesi
```

- ```
->Usus Numarasi: 122
->Kalkis Yeri: a
->Varis Yeri: c
->Kalkis Zamani: 12:45
->Normal Koltuk Sayisi: 5
->Luks Koltuk Sayisi: 9
```

Bu şekilde yazdıracak.

## 7) Bir yere olan uçuşların listelenmesi

- ```
->Varis yeri girin:
```

 varış yeri istenir.

```
->Varis yeri girin: c
->Ucus Bilgileri:
    ->Ucus Numarasi: 122
    ->Kalkis Yeri: a
    ->Varis Yeri: c
    ->Kalkis Zamani: 12:45
    ->Normal Koltuk Sayisi: 5
    ->Luks Koltuk Sayisi: 9
```

varış yeri girdikten sonra bu varış yerine uçuşlar var ise uçuşların bilgileri yazdıracak.

8) Bir yerden kalkan ucuslarin listelenmesi

->Kalkis yeri girin:

kalkış yeri istenir.

```
Bir yerden kalkan ucuslarin listelenmesi

->Kalkis yeri girin: a
->Ucus Bilgileri:
    ->Ucus Numarasi: 123
    ->Kalkis Yeri: a
    ->Varis Yeri: v
    ->Kalkis Zamani: 12:12
    ->Normal Koltuk Sayisi: 5

->Ucus Bilgileri:
    ->Ucus Numarasi: 122
    ->Kalkis Yeri: a
    ->Varis Yeri: c
    ->Kalkis Zamani: 12:45
    ->Normal Koltuk Sayisi: 5
    ->Luks Koltuk Sayisi: 9
```

kalkış yeri girdikten sonra bu kalkış yerine
uçuşlar var ise uçuşların bilgileri yazdıracak.

9) Bir yerden bir yere olan ucuslarin listelenmesi

->Kalkis yeri girin:

->Varis yeri girin:

```
Bir yerden bir yere olan ucuslarin listelenmesi

->Kalkis yeri girin: a
->Varis yeri girin: v
    ->Ucus Bilgileri:
        ->Ucus Numarasi: 123
        ->Kalkis Yeri: a
        ->Varis Yeri: v
        ->Kalkis Zamani: 12:12
        ->Normal Koltuk Sayisi: 5
```

Main.cpp

```
include <iostream>
```

```
#include <string>
```

```
#include "Ucus.h"
```

```
#include "Ucus_Numarasina_Gore_Ucus_Listesi.h"
```

```
#include "Kalkis_Yerine_Gore_Ucus_Listesi.h«
```

```
void L_icermeyen_yeniUcusEkle(Ucus_Numarasina_Gore_Ucus_Listesi& ucusN, Kalkis_Yerine_Gore_Ucus_Listesi& kalkisY);
```

```
void L_iceren_yeniUcusEkle(Ucus_Numarasina_Gore_Ucus_Listesi& ucusN, Kalkis_Yerine_Gore_Ucus_Listesi& kalkisY);
```

```
void ucuslariListele(const Ucus_Numarasina_Gore_Ucus_Listesi& ucusN, const Kalkis_Yerine_Gore_Ucus_Listesi& kalkisY);
```

```
Ucus_Numarasina_Gore_Ucus_Listesi ucusN;
```

Kalkis_Yerine_Gore_Ucus_Listesi kalkisY;

```
int main(){
```

[illegible]

```
cout<<"\t\t\t\t || MMS Havayolu Ucus Takip Sistemine hosgeldiniz\t\t ||\n";
```

[illegible]

```
cout<<"\n\n\n\n";
```

```
int secim;
```

```
int devametmek;
```

```
do {  
    cout << "\n1. Luks koltuk icermeyen yeni bir ucus ekle" << endl;  
    cout << "2. Luks koltuk iceren Yeni bir ucus ekle" << endl;  
    cout << "3. Bir ucusun luks koltuk sayisini guncellenmesi" << endl;  
    cout << "4. Bir ucusun iptal edilmesi" << endl;  
    cout << "5. Bir ucusun bilgilerini goruntulenmesi" << endl;  
    cout << "6. Luks koltuk iceren ucularin listelenmesi" << endl;  
    cout << "7. Bir yere olan ucuslarin listelenmesi" << endl;  
    cout << "8. Bir yerden kalkan ucuslarin listelenmesi" << endl;  
    cout << "9. Bir yerden bir yere olan ucuslarin listelenmesi" << endl;  
    cout << "\n";  
    cout << "yapacaginiz islem numarasi lutfen girin: ";  
    cin >> secim;  
    switch (secim) {  
        case 1:  
            L_icermeyen_yeniUcusEkle(ucusN, kalkisY);  
            break;  
        case 2:  
            L_iceren_yeniUcusEkle(ucusN, kalkisY);  
            break;  
        case 3:  
            ucusN.birUcusunLuksKoltukG();  
            break;
```



```
        case 4:
            ucusN.birUcusuSil();

            break;
    case 5:
        kalkisY.ucusBilgileriniGoruntule();

        break;
    case 6:
        ucusN.luks_koltuk_iceren_ucular();

        break;
    case 7:
        kalkisY.bir_yere_olan_ucus();

        break;
    case 8:
        kalkisY.bir_yerden_olan_ucuslar();

        break;
    case 9:
        kalkisY.bir_yerden_bir_yere_olan_ucuslar();

        break;
    default:
        cout << "\n\t\t\tGecersiz secim. Tekrar deneyin.\n" << endl;

        break;}

    cout<<endl<<"\nbaska bir islem yapmak ister misiniz? (1/0): ";
    cin>> devametmek;

    } while (devametmek != 0);

    return 0;

}
```

```
void L_icermeyen_yeniUcusEkle(Ucus_Numarasina_Gore_Ucus_Listesi& ucusN, Kalkis_Yerine_Gore_Ucus_Listesi& kalkisY){
```

```
    int ucusNumarasi;
```

```
    string kalkisYeri, varisYeri, kalkisZamani;
```

```
    int normalKoltukSayisi;
```

```
    cout << "\\nluks koltuk icermeyen Yeni bir ucus eklemesi\\n" << endl;
```

```
    cout << "\\t->Ucus numarasini girin: ";
```

```
    cin >> ucusNumarasi;
```

```
    if (ucusN.ucusVarMi(ucusNumarasi)) {
```

```
        cout << "\\n\\t\\t\\tBu numaraya sahip bir ucus zaten var.!!\\n" <<endl;
```

```
        return;
```

```
    }
```

```
    cout << "\\t->Kalkis yeri girin: ";
```

```
    cin.ignore();
```

```
    getline(cin, kalkisYeri);
```

```
    cout << "\\t->Varis yeri girin: ";
```

```
    getline(cin, varisYeri);
```

```
    cout << "\\t->Kalkis zamani girin: ";
```

```
    cin >> kalkisZamani;
```

```
    cout << "\\t->Normal koltuk sayisini girin: ";
```

```
    cin >> normalKoltukSayisi;
```

```
    Ucus* yeniUcus = new Ucus(ucusNumarasi, kalkisYeri, varisYeri, kalkisZamani, normalKoltukSayisi,0);
```

```
    ucusN.ucusEkle(yeniUcus);
```

```
    kalkisY.ucusEkle(yeniUcus);
```

```
    cout << "\\n\\t\\t\\tUcus basariyla eklendi...\\n" << endl;
```

```
}
```

```
void t_iceren_yeniUcusEkle(Ucus_Numarasina_Gore_Ucus_Listesi& ucusN, Kalkis_Yerine_Gore_Ucus_Listesi& kalkisY){

    int ucusNumarasi;

    string kalkisYeri, varisYeri, kalkisZamani;

    int normalKoltukSayisi,luksKoltukSayisi;


    cout << "\nluks koltuk iceren Yeni bir ucus eklemesi\n" << endl;

    cout << "\nt->Ucus numarasini girin: ";

    cin >> ucusNumarasi;


    if (ucusN.ucusVarMi(ucusNumarasi)) {

        cout << "\n\nt\nt\ntBu numaraya sahip bir ucus zaten var.\n" <<endl;

        return;

    }


    cout << "\nt->Kalkis yeri girin: ";

    cin.ignore();

    getline(cin, kalkisYeri);


    cout << "\nt->Varis yeri girin: ";

    getline(cin, varisYeri);


    cout << "\nt->Kalkis zamani girin: ";

    cin >> kalkisZamani;


    cout << "\nt->Normal koltuk sayisini girin: ";

    cin >> normalKoltukSayisi;


    cout << "\nt->Luks koltuk sayisini girin: ";

    cin >> luksKoltukSayisi;


    Ucus* yeniUcus = new Ucus(ucusNumarasi, kalkisYeri, varisYeri, kalkisZamani, normalKoltukSayisi,luksKoltukSayisi);

    ucusN.ucusEkle(yeniUcus);

    kalkisY.ucusEkle(yeniUcus);


    cout << "\n\nt\nt\ntUcus basariyla eklendi...\n" << endl;

}
```

Ucus.h

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class Ucus {
```

```
private:
```

```
    int ucusNumarasi;
```

```
    string kalkisYeri;
```

```
    string varisYeri;
```

```
    string kalkisZamani;
```

```
    int normalKoltukSayisi;
```

```
    int luksKoltukSayisi;
```

```
    Ucus* sonraki;
```

```
public:
```

```
    // Constructor
```

```
    Ucus(int numara, const string& kalkis, const string& varis, const string& zaman, int normalKoltuk, int luksKoltuk)
```

```
    : ucusNumarasi(numara), kalkisYeri(kalkis), varisYeri(varis), kalkisZamani(zaman),
```

```
      normalKoltukSayisi(normalKoltuk), luksKoltukSayisi(luksKoltuk), sonraki(nullptr) {}
```

```
    // Getter ve Setter fonksiyonları
```

```
    int getUcusNumarasi() const {
```

```
        return ucusNumarasi;
```

```
    }
```

```
    string getKalkisYeri() const {
```

```
        return kalkisYeri;
```

```
    }
```

```
    string getVarisYeri() const {
```

```
        return varisYeri;
```

```
    }
```

```

string getKalkisZamani() const {

    return kalkisZamani;

}

int getNormalKoltukSayisi() const {

    return normalKoltukSayisi;

}

int getLuksKoltukSayisi() const {

    return luksKoltukSayisi;

}

Ucus* getSonraki() const {

    return sonraki;

}

void setSonraki(Ucus* next) {

    sonraki = next;

}

void setLuksKoltukSayisi(int yeniSayi) {

    luksKoltukSayisi = yeniSayi;

}


// << operatorünün aşırı yüklenmesi

friend ostream& operator<<(ostream& os, const Ucus& ucus) {

    os << "Ucus Numarasi: " << ucus.ucusNumarasi << endl;

    os << "Kalkis Yeri: " << ucus.kalkisYeri << endl;

    os << "Varis Yeri: " << ucus.varisYeri << endl;

    os << "Kalkis Zamani: " << ucus.kalkisZamani << endl;

    os << "Normal Koltuk Sayisi: " << ucus.normalKoltukSayisi << endl;

    os << "Luks Koltuk Sayisi: " << ucus.luksKoltukSayisi << endl;

    return os;

}

};

```

Ucus_Numarasina_Gore_Ucus_Listesi.h

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class Ucus_Numarasina_Gore_Ucus_Listesi {
```

```
private:
```

```
    Ucus* bas;
```

```
public:
```

```
    // Constructor
```

```
    Ucus_Numarasina_Gore_Ucus_Listesi() : bas(nullptr) {}
```

```
    // Yeni uçupun eklenmesi
```

```
    void ucusEkle(Ucus* yeniUcus) {
```

```
        if (bas == nullptr) {
```

```
            bas = yeniUcus;
```

```
        } else {
```

```
            Ucus* temp = bas;
```

```
            Ucus* prev = nullptr;
```

```
            while (temp != nullptr && temp->getUcusNumarasi() < yeniUcus->getUcusNumarasi()) {
```

```
                prev = temp;
```

```
                temp = temp->getSonraki();
```

```
            }
```

```
            if (temp == nullptr) {
```

```
                prev->setSonraki(yeniUcus);
```

```
            } else if (temp->getUcusNumarasi() == yeniUcus->getUcusNumarasi()) {
```

```
                cout << "Bu numaraya sahip bir ucus zaten var." << endl;
```

```
            } else {
```

```
                if (prev == nullptr) {
```

```
                    yeniUcus->setSonraki(bas);
```

```
                    bas = yeniUcus;
```

```
                } else {
```

```
                    prev->setSonraki(yeniUcus);
```

```
                    yeniUcus->setSonraki(temp);
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
bool ucusVarMi(int ucusNumarasi) const {

    Ucus* current = bas;

    while (current != nullptr) {

        if (current->getUcusNumarasi() == ucusNumarasi) {

            return true;

        }

        current = current->getSonraki();

    }

    return false;

}

void birUcusunLuksKoltukG() {

    int ucusNumarasi, yeni_luks_koltuk_sayisi;

    Ucus* current = bas;

    cout << "\nBir ucusun luks koltuk sayisini guncellenmesi\n" << endl;

    cout << "\t->Ucus numarasi girin: ";

    cin >> ucusNumarasi;

    if (ucusVarMi(ucusNumarasi)) {

        cout << "\t->Yeni luks sayisini girin: ";

        cin >> yeni_luks_koltuk_sayisi;

        while (current != nullptr) {

            if (current->getUcusNumarasi() == ucusNumarasi) {

                current->setLuksKoltukSayisi(yeni_luks_koltuk_sayisi);

                break;

            }

            current = current->getSonraki();

        }

        cout << "\n\t\t\tLuks sayisi basariyle guncellendi.\n" << endl;

    }

    else {

        cout << "\n\t\t\tBu ucus numaraya ait bir ucus yoktur.\n" << endl;

    }

}
```

```
void birUcusuSil() {

    int ucusNumarasi;

    Ucus* current = bas;

    Ucus* prev = nullptr;

    cout << "\t->Ucus numarasi girin: ";

    cin >> ucusNumarasi;

    while (current != nullptr) {

        if (current->getUcusNumarasi() == ucusNumarasi) {

            if (prev != nullptr) {

                prev->setSonraki(current->getSonraki());

            } else {

                bas = current->getSonraki();

            }

            delete current;

            cout << "\n\t\t\tUcus basariyle silindi.\n" << endl;

            return; }

        prev = current;

        current = current->getSonraki(); }

    cout << "\n\t\t\tBu ucus numaraya ait bir ucus bulunamadi.\n" << endl; }

void luks_koltuk_iceren_ucular() {

    Ucus* current = bas;

    bool bulundu = false;

    cout << "\nLuks koltuk iceren uclarin listelenmesi" << endl;

    while (current != nullptr) {

        if (current->getLuksKoltukSayisi() > 0) {

            cout << "\n\t->Usus Numarasi: " << current->getUcusNumarasi() << "\n";

            cout << "\t->Kalkis Yeri: " << current->getKalkisYeri() << "\n";

            cout << "\t->Varis Yeri: " << current->getVarisYeri() << "\n";

            cout << "\t->Kalkis Zamani: " << current->getKalkisZamani() << "\n";

            cout << "\t->Normal Koltuk Sayisi: " << current->getNormalKoltukSayisi() << "\n";

            cout << "\t->Luks Koltuk Sayisi: " << current->getLuksKoltukSayisi() << "\n";

            bulundu = true; }

        current = current->getSonraki(); }

    if (!bulundu) {

        cout << "\n\t\t\tLuks koltuk iceren usus bulunamadi.\n"; } }

};
```


Kalkis_Yerine_Gore_Ucus_Listesi.h

```
#include <iostream>
```

```
#include <string>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
class Kalkis_Yerine_Gore_Ucus_Listesi {
```

```
private:
```

```
    struct Dugum {
```

```
        Ucus* ucus;
```

```
        Dugum* onceki;
```

```
        Dugum* sonraki;
```

```
        // Constructor
```

```
        Dugum(Ucus* u, Dugum* prev = nullptr, Dugum* next = nullptr)
```

```
            : ucus(u), onceki(prev), sonraki(next) {}
```

```
};
```

```
    Dugum* bas;
```

```
public:
```

```
    // Constructor
```

```
    Kalkis_Yerine_Gore_Ucus_Listesi() : bas(nullptr) {}
```

```
    // Yeni uçupun eklenmesi
```

```
    void ucusEkle(Ucus* yeniUcus) {
```

```
        Dugum* yeniDugum = new Dugum(yeniUcus);
```

```
        if (bas == nullptr) {
```

```
            bas = yeniDugum;
```

```
        } else {
```

```
Dugum* temp = bas;

Dugum* prev = nullptr;

while (temp != nullptr && (temp->ucus->getKalkisYeri() < yeniUcus->getKalkisYeri() ||

    (temp->ucus->getKalkisYeri() == yeniUcus->getKalkisYeri() && temp->ucus->getKalkisZamani() < yeniUcus->getKalkisZamani())) {

    prev = temp;

    temp = temp->sonraki;

}

if (temp == nullptr) {

    prev->sonraki = yeniDugum;

    yeniDugum->onceki = prev;

} else if (temp->ucus->getKalkisYeri() == yeniUcus->getKalkisYeri() && temp->ucus->getKalkisZamani() == yeniUcus->getKalkisZamani()) {

    cout << "Bu kalkis yerine sahip ve kalkis zamani olan bir ucus zaten var." << endl;

    delete yeniDugum;

} else {

    if (prev == nullptr) {

        yeniDugum->sonraki = bas;

        bas->onceki = yeniDugum;

        bas = yeniDugum;

    } else {

        prev->sonraki = yeniDugum;

        yeniDugum->onceki = prev;

        yeniDugum->sonraki = temp;

        temp->onceki = yeniDugum;

    }    }}}

void birUcusuSil(int y) {

    Dugum* current = bas;

    while (current != nullptr) {

        if (current->ucus->getUcusNumarasi() == y) {

            if (current->onceki != nullptr) {

                current->onceki->sonraki = current->sonraki;

            } else {

                bas = current->sonraki;

            }

        }

    }

}
```

```
if (current->sonraki != null || ptr) {

    current->sonraki->onanci = current->onanci;

}

delete current->ucus;

delete current;

return;

}

current = current->sonraki;

}

}

void ucusBilgileriGoruntule() const {

    int ucusNumarasi;

    cout<<"\nBir ucusun bilgilerini goruntulenmesi\n"<< endl;

    cout<<"\n>Ucus numarasini giriniz:";

    cin>>ucusNumarasi;

    if (bas == null || ptr) {

        cout<<"\n(!)Bu ucus numarasina ait bir ucus yoktur.\n"<< endl;

        return;

    }

    Dugum* temp = bas;

    while (temp != null || ptr) {

        if (temp->ucus->getUcusNumarasi() == ucusNumarasi) {

            if (temp->ucus->getLukotlukSayisi()>0) {

                cout<<"\nU>Ucus Bilgileri:"<< endl;

                cout<<"\nU>Ucus Numarası:"<<temp->ucus->getUcusNumarasi()<< endl;

                cout<<"\nU>U>Kalkis Yeri:"<<temp->ucus->getKalkisYeri()<< endl;

                cout<<"\nU>U>Varis Yeri:"<<temp->ucus->getVarisYeri()<< endl;

                cout<<"\nU>U>Kalkis Zamanı:"<<temp->ucus->getKalkisZamani()<< endl;

                cout<<"\nU>U>Normal Koltuk Sayisi:"<<temp->ucus->getNormalKoltukSayisi()<< endl;

                cout<<"\nU>U>Ekstra Koltuk Sayisi:"<<temp->ucus->getEkstraKoltukSayisi()<< endl;

                return; }

            else if (temp->ucus->getLukotlukSayisi()<1) {

                cout<<"\nU>U>Ucus Bilgileri:"<< endl;

                cout<<"\nU>U>Ucus Numarası:"<<temp->ucus->getUcusNumarasi()<< endl;

                cout<<"\nU>U>Kalkis Yeri:"<<temp->ucus->getKalkisYeri()<< endl;

                cout<<"\nU>U>Varis Yeri:"<<temp->ucus->getVarisYeri()<< endl;

                cout<<"\nU>U>Kalkis Zamanı:"<<temp->ucus->getKalkisZamani()<< endl;

                cout<<"\nU>U>Normal Koltuk Sayisi:"<<temp->ucus->getNormalKoltukSayisi()<< endl;

                return; }

            temp = temp->sonraki;

        }

    }

}
```

```
void bir_yere_olan_ucus(){
string varisyeri;

cout<< "\nBir yere olan ucuslarin listelenmesi\n"<< endl;

cout<< "\t->Varis yeri girin: ";

cin>>varisyeri;

if (bas == nullptr) {

    cout << "\n\t\t\tBu varis yerine ucus yoktur.\n" << endl;

    return;

}

Dugum* temp = bas;

while (temp != nullptr) {

    if (temp->ucus->getVarisYeri().compare(varisyeri) == 0) {

        if(temp->ucus->getLuksKoltukSayisi()>0){

            cout << "\t\t->Ucus Bilgileri:" << endl;

            cout << "\t\t\t->Ucus Numarasi: " << temp->ucus->getUcusNumarasi() << endl;

            cout << "\t\t\t->Kalkis Yeri: " << temp->ucus->getKalkisYeri() << endl;

            cout << "\t\t\t->Varis Yeri: " << temp->ucus->getVarisYeri() << endl;

            cout << "\t\t\t->Kalkis Zamani: " << temp->ucus->getKalkisZamani() << endl;

            cout << "\t\t\t->Normal Koltuk Sayisi: " << temp->ucus->getNormalKoltukSayisi() << endl;

            cout << "\t\t\t->Luks Koltuk Sayisi: " << temp->ucus->getLuksKoltukSayisi() << endl;

            cout<< "\n";

        }

        else if(temp->ucus->getLuksKoltukSayisi()<1){

            cout << "\t\t->Ucus Bilgileri:" << endl;

            cout << "\t\t\t->Ucus Numarasi: " << temp->ucus->getUcusNumarasi() << endl;

            cout << "\t\t\t->Kalkis Yeri: " << temp->ucus->getKalkisYeri() << endl;

            cout << "\t\t\t->Varis Yeri: " << temp->ucus->getVarisYeri() << endl;

            cout << "\t\t\t->Kalkis Zamani: " << temp->ucus->getKalkisZamani() << endl;

            cout << "\t\t\t->Normal Koltuk Sayisi: " << temp->ucus->getNormalKoltukSayisi() << endl;

            cout<< "\n";

        }

    }

    temp = temp->sonraki;

}}
```

```
cout << "\\t\\t->Varis Yeri: " << temp->ucus->getVarisYeri() << endl;

        cout << "\\t\\t->Kalkis Zamani: " << temp->ucus->getKalkisZamani() << endl;

        cout << "\\t\\t->Normal Koltuk Sayisi: " << temp->ucus->getNormalKoltukSayisi() << endl;

        cout<<"\\n";

    }

}

temp = temp->sonraki;

}

}

void bir_yerden_olan_ucuslar[] {

    string kalkisyeri;

    cout << "\\nBir yerden kalkan ucuslarin listelenmesi\\n" << endl;

    cout << "\\t->Kalkis yeri girin: ";

    cin >> kalkisyeri;

    if (bas == nullptr) {

        cout << "\\n\\t\\t\\tBu kalkis yerinden ucus yoktur.\\n" << endl;

        return;

    }

    Dugum* temp = bas;

    while (temp != nullptr) {

        if (temp->ucus->getKalkisYeri().compare(kalkisyeri) == 0) {

            if(temp->ucus->getLuksKoltukSayisi()>0){

                cout << "\\t\\t->Ucus Bilgileri:" << endl;

                cout << "\\t\\t->Ucus Numarasi: " << temp->ucus->getUcusNumarasi() << endl;

                cout << "\\t\\t->Kalkis Yeri: " << temp->ucus->getKalkisYeri() << endl;

                cout << "\\t\\t->Varis Yeri: " << temp->ucus->getVarisYeri() << endl;

                cout << "\\t\\t->Kalkis Zamani: " << temp->ucus->getKalkisZamani() << endl;

                cout << "\\t\\t->Normal Koltuk Sayisi: " << temp->ucus->getNormalKoltukSayisi() << endl;

                cout << "\\t\\t->Luks Koltuk Sayisi: " << temp->ucus->getLuksKoltukSayisi() << endl;

                cout<<"\\n";

            }

            else if(temp->ucus->getLuksKoltukSayisi()<1){

                cout << "\\t\\t->Ucus Bilgileri:" << endl;

                cout << "\\t\\t->Ucus Numarasi: " << temp->ucus->getUcusNumarasi() << endl;

                cout << "\\t\\t->Kalkis Yeri: " << temp->ucus->getKalkisYeri() << endl;
```

```
void bir_yerden_bir_yere_olan_ucuslar() {

    string kalkisyeri, varisyeri;

    cout << "\nBir yerden bir yere olan ucuslarin listelenmesi\n" << endl;

    cout << "\t->Kalkis yeri girin: ";
    cin >> kalkisyeri;

    cout << "\t->Varis yeri girin: ";
    cin >> varisyeri;

    if (bas == nullptr) {

        cout << "\n\t\t\tBu kalkis yerinden bu varis yere ucus yoktur.\n" << endl;

        return;

    }

    // Liste sonundan başa doğru ilerleyeceğimiz için, son düğümü bulmamız gerekiyor
    Dugum* sonDugum = bas;

    while (sonDugum->sonraki != nullptr) {

        sonDugum = sonDugum->sonraki;

    }

}
```

- // Son düğümden başlayarak geriye doğru ilerleyerek uçuşları yazdırıyoruz
- Dugum* temp = sonDugum;
- while (temp != nullptr) {
- if (temp->ucus->getKalkisYeri().compare(kalkisyeri) == 0 && temp->ucus->getVarisYeri().compare(varisyeri) == 0) {
- if (temp->ucus->getLuksKoltukSayisi() > 0) {
- cout << "\t\t->Ucus Bilgileri:" << endl;
- cout << "\t\t\t->Ucus Numarasi: " << temp->ucus->getUcusNumarasi() << endl;
- cout << "\t\t\t->Kalkis Yeri: " << temp->ucus->getKalkisYeri() << endl;
- cout << "\t\t\t->Varis Yeri: " << temp->ucus->getVarisYeri() << endl;
- cout << "\t\t\t->Kalkis Zamani: " << temp->ucus->getKalkisZamani() << endl;
- cout << "\t\t\t->Normal Koltuk Sayisi: " << temp->ucus->getNormalKoltukSayisi() << endl;
- cout << "\t\t\t->Luks Koltuk Sayisi: " << temp->ucus->getLuksKoltukSayisi() << endl;
- cout << "\n";
- } else if (temp->ucus->getLuksKoltukSayisi() < 1) {
- cout << "\t\t->Ucus Bilgileri:" << endl;
- cout << "\t\t\t->Ucus Numarasi: " << temp->ucus->getUcusNumarasi() << endl;
- cout << "\t\t\t->Kalkis Yeri: " << temp->ucus->getKalkisYeri() << endl;
- cout << "\t\t\t->Varis Yeri: " << temp->ucus->getVarisYeri() << endl;
- cout << "\t\t\t->Kalkis Zamani: " << temp->ucus->getKalkisZamani() << endl;
- cout << "\t\t\t->Normal Koltuk Sayisi: " << temp->ucus->getNormalKoltukSayisi() << endl;
- cout << "\n";
- }
- }
- temp = temp->onceki;
- }
- }
- };