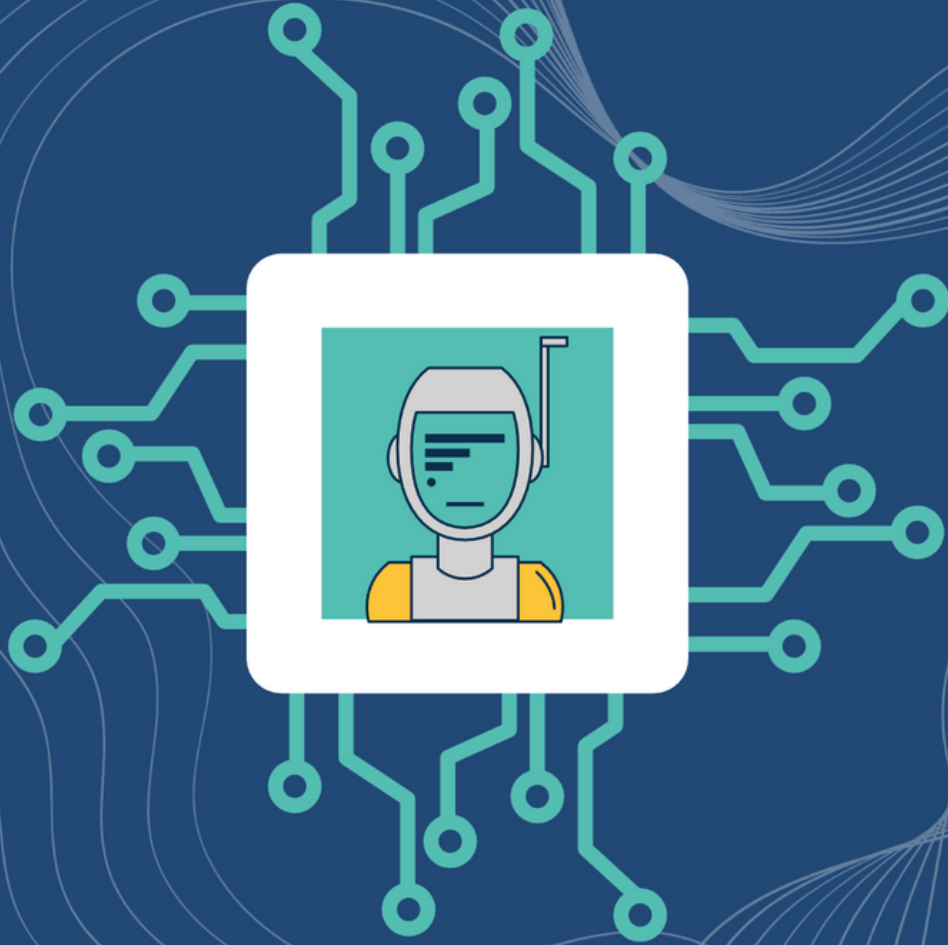


# NESNEYE YÖNELİK PROGRAMLAMA

## HAVAYOLU BİLET SATIŞ SİSTEMİ



ABDRAHMAN NACCAR - 220710008

MAHMUT EL-MUHAMMET - 220710078

TESLİM TARİHİ:05/07/2023

## **Programın amacı:**

-Programımızın çalışmasıyla kullanıcı aşağıdaki işlemleri yapabilecektir:

- 1)Sisteme yeni uçuş ekleme
- 2)Sistemde kayıtlı olan uçuş silme
- 3)Sistemde kayıtlı olan uçuşun kalkış saatinin düzeltmesi.
- 4)Bir yerden kalkan uçuşlar listelenmesi.
- 5) Bir yerden bir yere olan uçuşların bilgilerinin listelenmesi.
- 6) Bir uçuşun iptal edilmesi.
- 7)Bir uçuşa ilişkin bilet satın alması.
- 8) Bir yolcunun biletlerinin listelenmesi.
- 9) Bir uçuş için satılan bir biletin iptal edilmesi.

## Kullanıcının kataloğu:

```
Lutfen yapmak istediginiz islem numarasini giriniz:
1-Yeni bir ucusun eklenmesi
2-Bir ucusun kalkis zamaninin guncellemesi
3-Bir ucusa iliskin bilet satilmasi
4-Bir ucusun iptal edilmesi
5-Bir ucus icin satilan bir biletin iptal edilmesi
6-Bir yerden kalkan ucuslarin bilgilerinin listenmesi
7-Bir yerden yere olan ucuslarin bilgilerinin listelenmesi
8-Bir ucusun bilgilerinin ve o ucusa iliskin satilan biletlerin listelenmesi
9-Koltuk doluluk oranı %50'nin altında olan ucusların listelenmesi
10-Bir yolcunun biletlerinin listelenmesi
bitirmek icin (bitir) giriniz.
```

Burada yapmak istediginiz işlem numarasını girmeniz istenir.

### -Birinci işlem (yeni uçuş ekleme):

```
Ucus numarasini girinz:(1000 ile 9999 arasinda)
|
```

```
kalkis yeri giriniz:
|
```

```
varis yeri giriniz:
|
```

```
kalkis saati giriniz:
12:34|
```

kalkış saati bu formatla yazılması gerekir

```
koltuk kapasitesi giriniz:
|
```

```
baslangic fiyati giriniz:
|
```

## -İkinci işlem:(bir uçuşun kalkış zamanı güncellemesi)

```
Ucus numarasini girinz:
```

```
|
```

Güncellemek istediğiniz uçuşun numarası

```
Ucusun yeni kalkis saati giriniz:
```

```
13:00|
```

Saati bu formatla yazılması önemlidir.

## -Üçüncü işlem:(bilet alma)

```
Ucus numarasini girinz:
```

```
|
```

```
Satin alinacak bilet sayisini giriniz:
```

```
|
```

```
toplam fiyat: 56
```

```
Satin almak istiyor musunuz? (evet/hayir):
```

```
|
```

burada toplam ödenecek tutar yazılır  
ve onaylamak için evet giriniz

```
TC giriniz:
```

```
|
```

**-Dördüncü işlem:(bir uçuşun iptal edilmesi)**

```
Ucus numarasini giriniz:
```

```
|
```

Sistemden silmek istediğiniz uçuş numarasını giriniz

**-Beşinci işlem:(sistemden bilet silme)**

```
TC giriniz:
```

```
|
```

**-Altıncı işlem:(Bir yerden kalkan ucuslarin bilgilerinin listenmesi)**

```
Kalkis yerini giriniz:
```

```
|
```

**-Yedinci işlem:(Bir yerden yere olan ucuslarin bilgilerini listelenmesi)**

```
Kalkis yerini giriniz:
```

```
|
```

```
varis yerini giriniz:
```

```
|
```

**-Sekizinci işlem:**

Ucus numarasini giriniz:

|

**-Dokuzuncu işlem:**

Sistemde kayıtlı olup doluluk oranı 50 altında olan uçuşlar listelenir

**-Onuncu işlem:(Bir yolcunun biletlerinin listelenmesi)**

TC giriniz:

|

## Programcı kataloğu:

```
struct ucus
{
    int ucus_no;
    char kalkis_yeri[20];
    char varis_yeri[20];
    char zaman[5];
    int koltuk_kapasitesi;
    int bos_koltuk_sayisi;
    float baslangic_fiyati;
    ucus* Sonraki;
};
```

```
struct bilet
{
    char TC[11];
    int ucus_no;
    float bilet_fiyati;
    bilet* Sonraki;
};
```

Verileri depolamak için kullanılan structlar

```
ucus* ucuslar[9000];
int toplam_ucuslar = 0;
bilet* head_bilet = nullptr;
ucus* ucus_ky[9000];
int toplam_ucus_ky = 0;
```

## Programda kullanılan fonksiyonların prototipleri :

```
void Bir_ucusun_kalkis_zamaninin_guncellenmesi();
ucus* get_ucus(int ucus_numarasi);
void Bir_ucusa_iliskin_bilet_satilmasi();
int bilet_sayisi(int ucus_numarasi);
void insertAtend_bilet(int ucus_numarasi, float baslangic_fiyati);
void ucak_bileti iptali(int ucus_numarasi);
float fiyatBelirleme(float bko, float bbf);
void ucak_bileti iptali_TC();
void Yeni_bir_ucusun_eklenmesi();
void insert_ucus(int ucus_numarasi);
void insert_node_ucus_circular(ucus* new_node);
ucus* insertToEmptylist_ucus_circular(ucus* head_ucus, ucus* help);
ucus* insertAtend_ucus_circular(ucus* head_ucus, ucus* help);
bool check(ucus* head_ucus, int ucus_numarasi);
bool check_by_numarasi(int ucus_numarasi);
void bir_ucusun iptal edilmesi();
void delete_ucus(int ucus_numarasi);
void Bir_yerden_kalkan_ucuslari_bilgilerinin_listelenmesi();
void Bir_yerden_yere_olan_ucuslari_bilgilerini_listelenmesi();
void Bir_ucusun_bilgilerinin_ve_o_ucusa_iliskin_satilan_biletlerin_listelenmesi();
void bilet_gostermek_ucus_numarasi(int ucus_numarasi);
void Koltuk_doluluk_orani_50();
void Bir_yolcunun_biletlerinin_listelenmesi();
void sort_ucuslar();
```

```

void Yeni_bir_ucusun_eklenmesi() {
    bool Mevcut_yolculuk = false;
    int ucus_numarasi;
    do {
        cout << endl << "Ucus numarasini girinz:(1000 ile 9999 arasinda)" << endl;
        cin >> ucus_numarasi;
    } while (ucus_numarasi < 1000 || ucus_numarasi>9999);

    for (int i = 0; i < toplam_ucus_ky; i++)
    {
        if (check_by_numarasi(ucus_numarasi) == true) {
            Mevcut_yolculuk = true;
            break;
        }
    }
    if (Mevcut_yolculuk == false) {
        insert_ucus(ucus_numarasi);
    }
    else {
        cout << endl << "Numara daha once kayitliydi" <<endl<<endl;
    }
}

```

Bu fonksiyon yeni bir uçuş eklemek için tasarlandı

ilk önce kullanıcı gildiği numaraya ilişkin bir uçuş olmadığından kontroll edilir

ve (insert\_ucus) fonksiyonu çağırılır

```

void insert_ucus(int ucus_numarasi) {

    ucus* help = new ucus;
    ucuslar[toplam_ucuslar] = help;
    toplam_ucuslar++;
    help->ucus_no = ucus_numarasi;
    cout << endl << "kalkis yeri giriniz:" << endl;
    cin >> help->kalkis_yeri;
    cout << endl << "varis yeri giriniz:" << endl;
    cin >> help->varis_yeri;
    cout << endl << "kalkis saati giriniz:" << endl;
    cin >> help->zaman;
    cout << endl << "koltuk kapasitesi giriniz:" << endl;
    cin >> help->koltuk_kapasitesi;
    cout << endl << "baslangic fiyatı giriniz:" << endl;
    cin >> help->baslangic_fiyati;
    help->bos_koltuk_sayisi = 0;
    insert_node_ucus_circular(help);
    cout<<endl<<"Ucus basariyla eklendi."<<endl<<endl;
}

```

bu fonksiyonda kullanıcıdan bilgiler alınır ve bağlı listeye eklenir

Son olarak\*(insert\_node\_ucus\_circular) fonksiyonu çağırılır.



```

void insert_node_ucus_circular(ucus* new_node) {
    if (toplamlar_ucus_ky == 0) {
        ucus_ky[toplamlar_ucus_ky] = new_node;
        ucus_ky[toplamlar_ucus_ky]->Sonraki = ucus_ky[toplamlar_ucus_ky];
        toplamlar_ucus_ky++;
    }
    else {
        for (int i = 0; i < toplamlar_ucus_ky; i++)
        {
            if (ucus_ky[i]->kalkis_yeri[0] == new_node->kalkis_yeri[0]) {
                ucus_ky[i] = insertAtend_ucus_circular(ucus_ky[i], new_node);
                return;
            }
        }
        ucus_ky[toplamlar_ucus_ky] = new_node;
        toplamlar_ucus_ky++;
    }
    ucus* temp;
    for (int i = 0; i < toplamlar_ucus_ky - 1; i++)
    {
        for (int j = i + 1; j < toplamlar_ucus_ky; j++)
        {
            if (ucus_ky[i]->kalkis_yeri[0] > ucus_ky[j]->kalkis_yeri[0]) {
                temp = ucus_ky[i];
                ucus_ky[i] = ucus_ky[j];
                ucus_ky[j] = temp;
            }
        }
    }
}

```

Bu fonksiyonda kullanıcının girdiği uçuşların bilgilerine bakılır

ve aynı harfla başlayan uçuşların kalkış yerleri dairesel bağlı listeye eklenir

```

bool check_by_numarasi(int ucus_numarasi) {
    for (int i = 0; i < toplamlar_ucus_ky; i++)
    {
        if (ucuslar[i]->ucus_no == ucus_numarasi) {
            return true;
        }
    }
    return false;
}

```

Bu fonksiyon kullanıcının girdiği uçuş numarası önceden kayıtlı olmadığını kontrol etmek için tasarlandı.

diğer fonksiyonlar ve görevleri kod içinde açıklanmış

Zorlandığınız bir fonksiyon olursa irtibata geçiniz