

Temel Programlama I

Furkan : İSİM
KAYA : SOYİSİM
191216002 : NUMARA

Pelin : İSİM
GÜRKUT : SOYİSİM
191216009 : NUMARA

Öğrenci Otomasyonu

Mehmet Faik : İSİM
ÜVEN : SOYİSİM
191216010 : NUMARA

Kullanılan programlama dili: Python

Toplam satır sayısı: 240~

Arayüz: Konsol ekranı

Programımızın Amacı ve Yaptıkları

Yaptığımız programın amacı kullanıcıların rahatlıkla öğrencilerin notlarını girebilmesi, girilen notların değiştirilebilmesi, güncellenebilmesi ve öğrenci kaydının silinebilmesini sağlamaktadır. Yaptığımız programda yapılan işlemlerde alınabilecek tüm hatalarda kullanıcıya bilgi verilmektedir. İşlemlerin uzun olacağını düşünerek kullanıcının ekranı silebilmesi için de bir fonksiyon yazdık. Program açıldığında kullanıcının yapmak istediği işlemi seçmesi için ekranda seçenekler bulunmaktadır. Bu seçeneklerin hepsi birer numara tanımlıdır. Kullanıcı karşısına çıkan listeden yapmak istediği işlemin numarasını girerek seçer ve programı kullanmaya başlar.

Hoşgeldiniz. Aşağıdaki menüden işleminizi seçiniz.

- (1) Öğrencileri listele
- (2) Öğrenci ekle
- (3) Öğrenci sil
- (4) Öğrenci bilgilerini güncelle
- (0) Çıkış

İşlem: █

Kullanılan Fonksiyonlar:

Fonksiyon İsmi	Anlatım Sırası
Main	1
ListeleEkranı	2
TekOgrenciListeleEkranı	3
OgrenciGuncelleEkranı	4
OgrenciEkleEkranı	5
OgrenciSilEkranı	6
TumOgrencileriEkranayaBas	7
EkranıTemizle	8
EminlikSor	9
UygunGirisAl	10
OgrenciSil	11
SınıfOrtalamasınıDondur	12
ToplamPuanınıHesapla	13
YuzluktenHarfe	14
PuandanYuzluge	15
DigerSozlukleriGuncelle	16

1-) Main():

Yazılma amacı:

Programın akış şemasının ve işlemlerin daha anlaşılır bir şekilde yazılması için oluşturuldu.

Toplam satır sayısı: 27

Kabaca Algoritması:

- Ekranı temizle
- Menü metnini yaz
- Uygun bir giriş al
- İstenilen işlemin fonksiyonunu çağır
- Direkt resetlenmemesi enter tuşunu iste
- Başa dön

```
def Main():
    EkraniTemizle()
    print("Hoşgeldiniz. Aşağıdaki menüden işleminizi seçiniz.")
    print("""

(1) Öğrencileri listele
(2) Öğrenci ekle
(3) Öğrenci sil
(4) Öğrenci bilgilerini güncelle
(0) Çıkış

""")
    islem = str(UygunGirisAl("İşlem: ", (0,4), 1))
    EkraniTemizle()
    if islem=="1":
        ListeleEkrani()
    elif islem=="2":
        OgrenciEkleEkrani()
    elif islem=="3":
        OgrenciSilEkrani()
    elif islem=="4":
        OgrenciGuncelleEkrani()
    elif islem=="0":
        input("Hoşçakalın.")
        exit()
    else:
        print("Hatalı seçim yaptınız. Tekrar deneyiniz.")
        input("Ana menüye dönmek için enter tuşuna basınız...")
    Main()
```

2-) ListeleEkrani():

Yazılma amacı:

Listeleme türleri için menü sunmak ve kullanıcının seçtiği türde listeleme yapmak amacıyla oluşturuldu.

Toplam satır sayısı: 18

Kabaca Algoritması:

- Ekranı Temizle
- Menü Metnini Yaz
- Uygun seçim değerini al
- Seçime göre yönlendirme yap

```
def ListeleEkrani():
    EkraniTemizle()
    print("""

(1) Tek bir öğrenci listele
(2) Tüm öğrencileri listele
(0) Ana menüye dön

""")
    secim = str(UygunGirisAl("İşlem: ", (0,2), 1))

    if secim=="1":
        TekOgrenciListeleEkrani()
    elif secim=="2":
        EkraniTemizle()
        TumOgrencileriEkranBas()
    elif secim=="0":
        Main()
    else:
        print("Hatalı seçim yaptınız. Tekrar deneyiniz...")
```

3-) TekOgrenciListeleEkranı():

Yazılma amacı:

Kullanıcının istediği kişiyi görüntüleyip puan durumunu göstermek için oluşturuldu.

Toplam Satır sayısı: 8

Kabaca Algoritması:

- Ekranı Temizle
- Uygun bir öğrenci numarası al
- Öğrenci varmı yok mu kontrol et
- Yoksa uyarı ver, fonksiyondan çık
- Varsa öğrenci bilgilerini ekrana bas

```
166 def TekOgrenciListeleEkranı():
167     EkranıTemizle()
168     ogrenciNumarasi = str(UygunGirisAl("Öğrenci numarasını giriniz: ", (191216000,999999999), 9))
169     EkranıTemizle()
170     if ogrenciNumarasi not in ogrenciler:
171         print(ogrenciNumarasi + " nolu öğrenci mevcut değil.")
172         return
173     OgrenciBilgileriniEkranaBas(ogrenciNumarasi)
```

4-) OgrenciGuncelleEkranı():

Yazılma amacı:

Kullanıcının istediği kişiyi ve o kişinin hangi puan durumunu değiştirilmek isteniyorsa onun seçilip, üzerinde değişiklik yapmasını sağlamak için oluşturuldu.

Toplam Satır sayısı: 10

Kabaca Algoritması:

- Uygun bir öğrenci numarası al
- Öğrencinin olup olmadığını kontrol et
- Yoksa uyarı ver fonksiyondan çık
- Varsa
- Uygun bir not ismi al
- Uygun bir not yeni değeri al
- İşlem tamamlandı uyarısı göster
- Ana sözlüğü güncelle
- Yan sözlükleri güncelleme fonksiyonunu çağır

```
175 def OgrenciGuncelleEkranı():
176     ogrenciNumarasi = str(UygunGirisAl("Değiştirmek istediğiniz öğrenci numarasını giriniz: ", (191216000,999999999), 9))
177     if ogrenciNumarasi not in ogrenciler:
178         print(ogrenciNumarasi + " nolu öğrenci bulunmuyor.")
179         return
180     degistirilmekIstenenNot = str(UygunGirisAl((str(list(puanlamalar.keys())) + " arasından hangi notu değiştirmek istersiniz? "),
181         list(puanlamalar.keys()), ""))
182     EkranıTemizle()
183     yeniDegeri = UygunGirisAl((degistirilmekIstenenNot + " notunun yeni değerini ne yapmak istersiniz?: "),
184         [0, puanlamalar[degistirilmekIstenenNot]], "")
185     print("Not başarıyla değiştirildi.")
186     ogrenciler[ogrenciNumarasi][degistirilmekIstenenNot] = int(yeniDegeri)
187     DigerSozlukleriGuncelle(ogrenciNumarasi)
```

5-) OgrenciEkleEkranı():

Yazılma amacı:

Kullanıcının öğrenci ekleyebilmesi amacıyla oluşturuldu.

Toplam Satır sayısı: 13

Kabaca Algoritması:

- Eklenen notların ve öğrencinin tutulacağı sözlükleri oluştur
- Uygun bir öğrenci numarası iste
- Öğrenci var mı kontrol et
- Varsa uyarı ver fonksiyonu bitir
- Yoksa sırayla tüm notları kullanıcıdan iste
- Tek bir öğrenci için sözlük yapısını oluştur
- Ana sözlüğe bu öğrenciyi ekle
- Diğer sözlükleri güncelle fonksiyonunu çağır
- Eklendi uyarısını kullanıcıya ver

```
151 def OgrenciEkleEkranı():
152     notlar={}
153     ogrenci={}
154     ogrenciNumarasi = str(UygunGirisAl("Öğrenci numarasını giriniz: ", (191216000,999999999), 9))
155     if ogrenciNumarasi in ogrenciler:
156         print(ogrenciNumarasi + " nolu öğrenci zaten mevcut.")
157         return
158     for i in puanlamalar:
159         notlar[i] = UygunGirisAl((i + " Notunu giriniz: "), (0, puanlamalar[i]), "")
160     ogrenci[ogrenciNumarasi] = notlar
161     ogrenciler.update(ogrenci)
162     DigerSozlukleriGuncelle(ogrenciNumarasi)
163     EkranıTemizle()
164     print(ogrenciNumarasi + " nolu öğrenci başarıyla eklendi.")
```

6-) OgrenciSilEkranı():

Yazılma amacı:

Kullanıcının istediği öğrenciyi silmesini sağlamak için oluşturuldu. Bu işlem öğrencinin numarası alınarak gerçekleştirilmektedir. Öğrenci numarası yok ise uyarı vermektedir.

Toplam Satır sayısı: 6

Kabaca Algoritması:

- Ekranı Temizle
- Uygun bir öğrenci numarası al
- Öğrenci var mı kontrol et
- Öğrenci varsa
 - emin misiniz sor ve evetse öğrenci numarasıyla ogrencisil fonksiyonunu çalıştır
- Yoksa
 - mevcut değil uyarısını ver

```
def OgrenciSilEkranı():
    EkranıTemizle()
    silinecekid = str(UygunGirisAl("Silmek istediğiniz öğrencinin numarasını giriniz: ", (191216000,999999999), 9))
    if silinecekid in ogrenciler:
        EminlikSor((silinecekid + " nolu öğrenciyi silmek istediğinize emin misiniz? e/h:"), OgrenciSil, silinecekid)
    else:
        print(silinecekid + " nolu öğrenci mevcut değil.")
```

7-) TumOgrencileriEkранаBas():

Yazılma amacı:

Kullanıcının sözlükte bulunan tüm öğrencileri ve notlarını ekrana vermesini sağlamak için oluşturuldu.

```
def TumOgrencileriEkранаBas():  
    for ogrenciNumarasi in ogrenciler:  
        OgrenciBilgileriniEkранаBas(ogrenciNumarasi)  
    print("Sınıf ortalaması: " + str(round(SınıfOrtalamasiniDondur(),2)))
```

Toplam Satır sayısı:4

Kabaca Algoritması:

- Olan tüm öğrencilerin idlerini ogrencibilgileriniekranabas fonksiyonuna gönder
- Sınıf ortalamasını ekrana bas

8-) EkranıTemizle():

Yazılma amacı:

Programın çalıştığı işletim sistemine göre bağımsız olarak gerekli yerlerde ekranı temizleme amacıyla oluşturuldu.

```
115 def EkranıTemizle():  
116     # cls komutu ubuntu'da clear komutu windowsta çalışmadığı için  
117     # programın çalıştığı işletim sistemini kontrol edip ona göre komut çalıştırıyoruz.  
118     if "windows" in platform.platform().lower():  
119         os.system("cls")  
120     if "linux" in platform.platform().lower():  
121         os.system("clear")
```

Toplam Satır sayısı: 6

Kabaca Algoritması:

- Windows ise
 - cls komutunu çalıştırıp ekranı temizle
- Linux sistemlerindense
 - clear komutunu çalıştırıp ekranı temizle

9-) EminlikSor():

Yazılma amacı:

Programın değişik yerlerinde evet veya hayır şeklinde onayla işlem yapmak amacıyla oluşturuldu.

Toplam Satır sayısı: 11

Kabaca Algoritması:

- Doğru değer gelene kadar tekrar etmesi için sonsuz döngüye sok
- Ekranı temizle
- Ekrana yazılacak değerini yaz ve dönüş al
- Gelen değer e ise fonksiyona verilmiş efonkunu yine fonksiyona verilmiş parametreyle çağır ve döngüyü bitir
- h ise vazgeçildi yaz döngüyü bitir
- Bunlar dışında bir değerse hatalı giriş yapıldı de ve başa al

```
def EminlikSor(ekranaYazilacak, eFonk, eFonkParametresi):
    while True:
        EkranıTemizle()
        eminlik = input(ekranaYazilacak)
        if eminlik == "e":
            eFonk(eFonkParametresi)
            break
        elif eminlik == "h":
            print("Vazgeçildi")
            break
        else:
            input("Hatalı bir giriş yaptınız. Tekrar denemek için enter tuşuna basınız.")
```

10-) UygunGirisAl():

Yazılma amacı:

Fonksiyonumuzun amacı kullanıcının girdiği değerlerin seçtikleri işleme uygun olup olmadığını test etmek için oluşturuldu.

Toplam Satır sayısı: 19

Kabaca Algoritması:

- Doğru değer gelene kadar dönmesi amacıyla sonsuz döngüye gir
- Parametrede verilmiş yazıyla input iste
- Parametrelerin sağlanıp sağlanmadığını kontrol et
- Sağlanıyorsa
- girilen değeri geri döndür
- Sağlanmıyorsa
- Sağlanmayan koşulun hata mesajını göster tekrar input iste

```
79 def UygunGirisAl(ekranaYazilacak, degerAraligi, karakterSayisi):
80     while True:
81         girilen = input(ekranaYazilacak)
82         if type(degerAraligi) == tuple:
83             if girilen.isdigit():
84                 if len(girilen) == karakterSayisi or karakterSayisi == "":
85                     girilen = int(girilen)
86                     if degerAraligi[0] <= girilen <= degerAraligi[1]:
87                         return girilen
88                 else:
89                     print("Girdiğiniz değer " + str(degerAraligi[0]) + "'dan küçük " + str(degerAraligi[1]) + "'den büyük olamaz.")
90             else:
91                 print("Girdiğiniz değer " + str(karakterSayisi) + " karakter içermelidir.")
92         else:
93             print("Girdiğiniz değer sayı olmalıdır.")
94         elif type(degerAraligi) == list:
95             for i in degerAraligi:
96                 if i.lower() == girilen.lower():
97                     return i
98             print("Girdiğiniz değer şunlardan biri olmalıdır: " + str(degerAraligi))
```


11-) OgresnciSil():

Yazılma amacı:

Öğrenci idsi ile öğrenciyi sistemden tamamen kaldırabilmek amacıyla oluşturuldu.

Toplam Satır sayısı: 8

Kabaca Algoritması:

- Ekranı temizle
- Öğrenci var mı kontrol et
- varsa
 - ogrenciler sözlüğünden öğrenciyi kaldır
 - diğer sözlüklerden de kaldırmak için diğer sözlükleri güncelle fonksiyonunu çağır
- yoksa
 - öğrencinin bulunmadığını bildir

```
68 def OgresnciSil(ogrenciId):
69     EkranıTemizle()
70     if ogrenciId in ogrenciler:
71         del ogrenciler[ogrenciId]
72         DigerSozlukleriGuncelle(ogrenciId)
73         print(ogrenciId + " nolu öğrenci başarıyla silindi.")
74     else:
75         print(ogrenciId + " nolu öğrenci bulunmuyor.")
```

12-) SınıfOrtalamasiniDondur():

Yazılma amacı:

Tüm öğrencileri listeleme ekranının en aşağısında sınıf ortalamasını göstermek amacıyla oluşturuldu.

Toplam Satır sayısı: 5

Kabaca Algoritması:

- Toplamı 0 a eşitle
- yuzlukNotlar listesindeki her bir öğrenciyi dolaş
- Öğrencinin puanını toplam değişkenine ekle
- toplam puanı öğrenci sayısına böl ve sonucu döndür

```
62 def SınıfOrtalamasiniDondur():
63     toplam=0
64     for ogrenci in yuzlukNotlar:
65         toplam = toplam + yuzlukNotlar[ogrenci]
66     return toplam/len(yuzlukNotlar)
```


13-) ToplamPuaniniHesapla():

Yazılma amacı:

Fonksiyonda girilen öğrencinin toplam puanını hesaplayıp döndürmesi amacıyla oluşturuldu.

Toplam Satır sayısı: 5

Kabaca Algoritması:

- ToplamPuanı 0 ile başlat
- Her bir puan türü için öğrenciden notları sırayla al
- Toplam puana ekle
- Toplam puanı döndür

```
def ToplamPuaniniHesapla(ogrenciId):  
    toplamPuan = 0  
    for notlar in puanlamalar:  
        toplamPuan += ogrenciler[ogrenciId][notlar]  
    return toplamPuan
```

14-) YuzluktenHarfe():

Yazılma amacı:

Fonksiyonun içine giren yüzlük sistemdeki notun harf karşılığını döndürmek için oluşturuldu.

Toplam Satır sayısı: 5

Kabaca Algoritması:

- Aralıkları dolaş
- Fonksiyonun içine verilmiş sayısal not aralıkta mı kontrol et
- Aralıktaysa
 - aralığın ismini döndür
- Hiç bir aralıkta değilse hatalı değer uyarısı ver

```
ortKarsiliklari = {"AA" : (90,100), "BA" : (80,89), "BB" : (70,79), "CB" : (60,69),  
                  "CC" : (50,59), "DC" : (40,49), "DD" : (30,39), "FF" : (0,19)}  
  
def YuzluktenHarfe(sayisalNot):  
    for ortKey in ortKarsiliklari:  
        if ortKarsiliklari[ortKey][0] <= sayisalNot <= ortKarsiliklari[ortKey][1]:  
            return ortKey  
    return "Hatalı değer."
```

15-) PuandanYuzluge():

Yazılma amacı:

Puanı hesaplanmış olan öğrencilerin, puanlarını yüzlük sisteme dönüştürmek için oluşturuldu.

Toplam Satır sayısı: 2

Kabaca Algoritması:

- Gelen değerin %5 ini döndür

```
36 def PuandanYuzluge(puan):  
37     return (puan/100)*5
```

16-)DigerSozlukleriGuncelle():

Yazılma amacı:

Kullanıcı bir kütüphaneden bir kişiyi sildiğinde, eklediğinde ya da güncellediğinde diğerlerini de güncellemek için oluşturuldu.

Toplam Satır sayısı: 8

Kabaca Algoritması:

- Öğrenci var mı kontrol et
- Varsa
 - En yeni notlarını hesapla ve tüm sözlükleri güncelle
- Yoksa
 - O öğrenciyi tüm sözlüklerden kaldır

```
24 def DigerSozlukleriGuncelle(ogrenciId):  
25     if ogrenciId in ogrenciler:  
26         toplamPuanlar[ogrenciId] = ToplamPuaniniHesapla(ogrenciId)  
27         yuzlukNotlar[ogrenciId] = PuandanYuzluge(toplamPuanlar[ogrenciId])  
28         harfNotlari[ogrenciId] = YuzluktenHarfe(yuzlukNotlar[ogrenciId])  
29     else:  
30         del toplamPuanlar[ogrenciId]  
31         del yuzlukNotlar[ogrenciId]  
32         del harfNotlari[ogrenciId]
```